

1N-39

~~55563~~

p- 116

Structural Design Using Equilibrium Programming Formulations

Stephen Joseph Scotti
Langley Research Center, Hampton, Virginia

(NASA-TM-110175) STRUCTURAL DESIGN
USING EQUILIBRIUM PROGRAMMING
FORMULATIONS (NASA. Langley
Research Center) 116 p

N95-30682

Unclass

G3/39 0055563

June 1995

National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia 23681-0001

Abstract

Solutions to ever-larger structural optimization problems are desired. However, computational resources are strained to meet this need. New methods will be required to solve ever larger problems. The present approaches to solving large-scale problems involve approximations for the constraints of structural optimization problems and/or decomposition of the problem into multiple subproblems that can be solved in parallel. An area of game theory, equilibrium programming (also known as non-cooperative game theory), can be used to unify these existing approaches from a theoretical point of view (considering the existence and optimality of solutions), and be used as a framework for the development of new methods for solving large-scale optimization problems. Equilibrium programming theory is described, and existing design techniques such as fully stressed design and constraint approximations are shown to fit within its framework. Two new structural design formulations are also derived. The first new formulation is another approximation technique which is a general updating scheme for the sensitivity derivatives of design constraints. The second new formulation uses a substructure-based decomposition of the structure for analysis and sensitivity calculations. The new formulations are utilized for problems ranging from a simple ten-bar truss to a 348 design-variable optimization of a high-speed civil transport. Significant computational benefits of the new formulations compared with a conventional method are demonstrated.

Acknowledgements

This paper is essentially a dissertation submitted to the faculty of the School of Engineering and Applied Science of The George Washington University in partial satisfaction of the requirements for the degree of Doctor of Science. I wish to express my sincere gratitude to my advisors, Dr. Raphael Haftka, and Dr. John Whitesides. I also would like to thank my family, especially my wife Joan, and children Brian, Sarah, Shea, and Samantha for their patience during my long tenure as a graduate student, and for their acceptance of the demands it placed on my time. Mr. Carl Martin was of great assistance in development of the SLP runstream used in the example problems, and in setting up and running some of the example problems. The contribution of the Boeing Company in providing the vehicle example problem is appreciated. The support and encouragement of the NASA Langley Research Center, in particular my branch head Dr. Charles Camarda and my former branch heads Dr. James Starnes and Dr. Donald Rummier, is gratefully acknowledged.

Table of Contents

Abstract	i
Acknowledgements	ii
Table of Contents	iii
List of Tables	v
List of Figures	vi
Nomenclature	viii
Chapter 1. Introduction	1
1.1 Background on Structural Synthesis	1
1.2 Scope of Present Study	8
Chapter 2. Equilibrium Programming Background	13
2.1 Equilibrium Programming Background	13
2.2 Features of Equilibrium Programming Problems	15
2.3 Existence of an Equilibrium Point	16
2.4 Some Equilibrium Point Properties	18
2.5 Solution Methodologies	20
Chapter 3. Development of Equilibrium Programming Structural- Design Formulations	22
3.1 Optimal Structural-Sizing Problem Statement	24
3.2 Structural-Design Formulation 1	26
3.3 Structural-Design Formulation 2	31
3.4 Structural-Design Formulation 3	34
3.4.1 Structural-Design Formulation Using Updated Sensitivity Deriva- tives	36

3.5	Design Formulation 4	38
3.5.1	Derivation of the Substructure Analysis Method	40
3.5.2	Derivation of EP Design Formulation 4	47
Chapter 4. Numerical Results for Equilibrium Programming Structural- Design Formulations		57
4.1	Implementation of Numerical Algorithms	57
4.1.1	General Implementation Issues	58
4.1.2	Implementation Issues for Design Formulation 3	61
4.1.3	Implementation Issues for Design Formulation 4	63
4.2	Description of the Example Problems	64
4.2.1	Ten-Bar-Truss Example Problem	65
4.2.2	High-Speed Civil Transport Wing Example Problem	66
4.2.3	High-Speed Transport Vehicle Example Problem	68
4.2.4	Transmission Tower Example Problem	69
4.3	Results for the Example Problems	71
4.3.1	Ten-Bar-Truss Example Problem Results	71
4.3.2	High-Speed Civil Transport Wing Example Problem Results	77
4.3.3	High-Speed Transport Vehicle Example Problem Results	82
4.3.4	Transmission Tower Example Problem Results	87
Chapter 5. Concluding Remarks		91
Bibliography		95

List of Tables

4.1	Comparison of normalized CPU timing results for the transmission tower example using EP structural-design formulation 4 with two and four substructures, and using a conventional SLP approach for the entire structure.	89
-----	--	----

List of Figures

3.1	Design space for a rod constrained between two rigid walls undergoing a temperature change.	31
3.2	Example wing structural finite element model decomposed into two substructures.	42
4.1	Flowchart of the solution procedure for structural design formulation 3.	62
4.2	Schematic of geometry, loads, and displacement limits for the ten-bar-truss example problem.	66
4.3	Finite element model of high-speed civil transport wing with upper cover panels removed.	67
4.4	Half-symmetric, finite element model of the high-speed transport vehicle.	69
4.5	Schematic of the transmission tower example geometry, loading, and division into substructures.	70
4.6	Comparison of penalized objective function iteration histories for the ten-bar truss with different initial move-limit-control factors using EP structural-design formulation 3, and using a conventional SLP approach.	73
4.7	Comparison of objective function iteration histories for the ten-bar truss with different initial move-limit-control factors using EP structural-design formulation 3, and using a conventional SLP approach.	75
4.8	Comparison of iteration histories for displacement constraint on δ_1 for the ten-bar truss with different initial move-limit-control factors using EP structural-design formulation 3, and using a conventional SLP approach.	75
4.9	Comparison of penalized objective function iteration histories using a CPU time ordinate for the ten-bar truss with different initial move-limit-control factors using EP structural-design formulation 3, and using a conventional SLP approach.	76
4.10	Comparison of penalized objective function iteration history for the high-speed civil transport wing with different initial move-limit-control factors using EP structural-design formulation 3, and using a conventional SLP approach.	78

4.11	Objective function and wing tip displacement constraint iteration histories for the high-speed civil transport wing using EP structural-design formulation 3, and using a conventional SLP approach.	81
4.12	Comparison of penalized objective function iteration history using a CPU time ordinate for the high-speed civil transport wing with different initial move-limit-control factors using EP structural-design formulation 3, and using a conventional SLP approach. .	81
4.13	Comparison of penalized objective function iteration history for the high-speed transport vehicle using EP structural-design formulation 3, using modified EP structural-design formulation 3, and using a conventional SLP approach.	83
4.14	Comparison of objective function and most critical stress constraint iteration history for the high-speed transport vehicle using EP structural-design formulation 3, using modified EP structural-design formulation 3, and using a conventional SLP approach. .	86
4.15	Comparison of penalized objective function iteration history using a CPU time ordinate for the high-speed transport vehicle using EP structural-design formulation 3, using modified EP structural-design formulation 3, and using a conventional SLP approach.	86
4.16	Comparison of objective function and most critical stress constraint iteration history for the transmission tower example using EP structural-design formulation 4 with two and four substructures, and using a conventional SLP approach for the entire structure.	88

Nomenclature

Column vectors are generally denoted by lower case symbols typed in boldface. The notation $\partial f / \partial \mathbf{a}$ for an arbitrary scalar f and a vector $\mathbf{a} = (a_1, \dots, a_n)^T$ denotes the gradient of scalar f with respect to vector \mathbf{a} . This gradient is expressed as a row vector (i.e., $\partial f / \partial \mathbf{a} = (\partial f / \partial a_1, \dots, \partial f / \partial a_n)$). The notation $\partial \mathbf{f} / \partial \mathbf{a}$ for arbitrary vectors \mathbf{a} and $\mathbf{f} = (f_1, \dots, f_m)^T$ denotes the gradient of vector \mathbf{f} with respect to vector \mathbf{a} . This gradient is expressed as an $m \times n$ matrix in which the component in column j of row i is given by $\partial f_i / \partial a_j$. The notation $f_{i,j}$ denotes the j^{th} component of the vector \mathbf{f}_i .

\mathbf{a}	update vector
A	cross-sectional area, sometimes (in formulation 4) the set of 4-tuples used to define compatibility of substructures
\mathbf{A}	vector of cross-sectional areas
\mathbf{B}_j	matrices used to define compatible displacements at substructure interfaces
\mathbf{g}	vector of inequality constraint functions
$\mathbf{e}_j^{\mathbf{v}}$	unit vector with length of vector \mathbf{v} and having unity for the j^{th} entry
f	objective function in nonlinear programming
\mathbf{F}	vector of external nodal forces
\mathbf{h}	vector of equality constraint functions
k	constant of proportionality
K	penalty coefficient used in penalized objective function P

$\mathbf{K}, \tilde{\mathbf{K}}$	global stiffness matrix for a structural-response problem, augmented global stiffness matrix for a structural-response subproblem (see equations (3.28) and (3.34))
m	number of substructures that would have rigid-body motions if separated from the rest of the structure, used in formulation 4
M	number of EP subproblems, sometimes number of substructures in formulation 4
\mathbf{M}	matrix defining the substructure coordination problem (see equations (3.29) and (3.36))
n	number of displacement degrees of freedom
\mathbf{N}	vector of stress resultants within every element of the structure
P	penalized objective function (see equation (4.2))
\mathbf{R}	matrix containing substructure rigid-body modes
\mathbf{u}	vector of nodal displacements from a structural-response problem
\mathbf{U}	vector of nodal displacements orthogonal to rigid-body modes
\mathbf{v}	vector of design variables for nonlinear programming, or vector of structural-sizing design variables in equilibrium programming structural-sizing subproblems
w_k	weights used in update subproblem (see statement (3.12))
W	weight of a structure
$\mathbf{x}, \mathbf{x}_i, \bar{\mathbf{x}}_i$	vectors of all design variables for equilibrium programming, design variables of subproblem i , and all design variables except those from subproblem i , respectively
$\boldsymbol{\alpha}$	vector of rigid-body mode amplitudes
δ	displacement value
$\Delta f, \Delta v_k$	difference quantities used in update equation (3.13)
ΔT	temperature rise

λ	vector of Lagrange multipliers for inequality constraints, sometimes Lagrange multipliers for constraint on compatible substructure interface displacements
μ	vector of Lagrange multipliers for equality constraints, sometimes vector of Lagrange multipliers for constraint on orthogonality of \mathbf{u} nodal displacements with rigid-body modes
σ	vector of stresses within every element of a structure

Subscripts and Subscripts:

A	denotes an approximation of a function
i	denotes i 'th equilibrium programming subproblem
I	denotes interface quantities in design formulation 4
l, u	denotes lower and upper bounds, used to denote move limits
max	maximum
min	minimum
new, old	new and old values, respectively; used in defining update subproblem (3.12)
N	denotes functional dependence on stress resultants
s	side constraints
T	transpose of a vector or matrix
\mathbf{u}	denotes displacement constraints, or functional dependence on displacements
σ	denotes stress and local buckling constraints, or functional dependence on stresses
$*$	denotes an equilibrium programming solution (i.e., the equilibrium point), an optimal solution, active constraints, or Lagrange multipliers corresponding to active constraints

Chapter 1

Introduction

Solutions to ever-larger structural optimization problems are needed as structural optimization procedures are applied to solve practical design problems and the analysis models for these structures grow increasingly detailed and complex. Current applications of structural optimization to large-scale design problems require significant computational resources. New structural optimization methods are required to solve ever-larger problems. The use of a generalized theoretical framework for the development of structural optimization methods is investigated in the present dissertation.

1.1 Background on Structural Synthesis

Nonlinear mathematical programming (NLP) is now extensively used for a type of optimal structural design that is called *structural sizing*. The solution to a NLP-based, structural-sizing design problem minimizes an objective function, such as weight, and satisfies a set of design constraints, such as minimum gauge and local stress limits. This solution is obtained varying a set of structural-sizing design variables, such as skin thicknesses and stiffener heights, within their allowable ranges. The original usage of NLP for structural sizing coupled with finite element structural analyses was due to Schmit in 1960 (see [50] for a

description of the historical developments). Schmit called his resulting method *structural synthesis*. The structural synthesis method had several advantages over the state of the art at that time (i.e., fully stressed design methods and simultaneous failure mode methods). These advantages include such features as: 1) limits on member sizes were easily incorporated; 2) a true optimum design could be found subject to multiple load cases; 3) numerous failure modes could be considered simultaneously; and 4) objective functions other than weight were possible. The disadvantage of the method was the large demand placed on computational resources.

In the original structural synthesis approach, the structural analysis required to compute the design constraints was subordinate to the optimization algorithm, and a structural analysis was performed for every change in the values of the structural-sizing design variables in the design process. Thus, the process was computationally expensive. Several methods for overcoming this disadvantage such as the use of approximation techniques, reduction techniques, and decomposition techniques were presaged as early as 1971 by Pope and Schmit [40].

Approximation techniques are utilized to loosen the coupling of the structural analysis from the structural optimization procedure. Computationally efficient, approximate models for the dependence of the structural response and the constraint functions on the design variables are developed from the structural analysis and the sensitivity derivatives of the response. These approximate models are used in place of the full structural analysis and constraint evaluation in the structural optimization procedure. After the structural optimization procedure converges, the approximate models for the structural re-

sponse are reformulated using the resulting set of design variables, and the cycle is repeated until the variation of results between cycles is below a predefined tolerance. The accuracy of using linear Taylor series approximate analyses was evaluated by Storaasli and Sobieszczanski in [63] using a finite element model of a subsonic transport fuselage section. However, Schmit and Farshi defined the approximation concepts commonly used today in [45], and the further development of these concepts by Schmit and Miura [48] significantly improved the computational efficiency of NLP structural-sizing design methods. Many of the numerous approximation techniques that have been developed are summarized in [2]. Thus, approximation-based NLP structural-sizing design methods are now well developed, and are implemented in several commercial structural analysis codes.

Reduction techniques are utilized to reduce the size of a structural optimization problem to make it more tractable. In one approach, the original set of design variables is replaced by a sum of fixed basis vectors for the design variables (chosen by the user) multiplied by undetermined coefficients. If the number of basis vectors is smaller than their length, these coefficients form a new set of design variables that is smaller than the original set of design variables (see [39]). However, the best choice of basis vectors is not known *a priori*, and the quality of the resulting design depends on the user's skill in choosing good basis vectors. In another reduction technique, the constraint functions are combined into one or more *cumulative constraint* functions. This approach can significantly reduce the number of constraints. Cumulative constraints may make the adjoint sensitivity method more attractive by reducing the number of constraints to a number significantly below the number of design variables ([19],

p. 267), and they are also utilized in some of the decomposition methods described subsequently. The Kreisselmeier-Steinhauser (K-S) function described in [27] is one example of a cumulative constraint function.

Another approach that has been investigated to improve the computational efficiency of structural sizing is the decomposition of the problem into a number of smaller subproblems that can be analyzed in parallel. An excellent summary of the different ways decomposition techniques are utilized to solve engineering design problems is given by Barthelemy [5] where decomposition methods are classified based on how the design variables are coupled through the problem constraints. This classification technique is similar to the method for classifying large linear programs ([29], pp. 117 - 122). Decomposition techniques become computationally advantageous if the overall problem can be reformulated into subproblems having sets of design variables and constraints that are nearly disjoint. In practice, there is usually some coupling between the subproblems. If some design variables are common to the constraints of the different subproblems, they are called *coupling variables*. If some constraints depend on design variables from more than one subproblem, they are called *coupling constraints*. In linear programming, problems with coupling variables, but without coupling constraints, are called *angular*, and problems with coupling constraints, but no coupling variables, are called *dual-angular*.

Some of the earliest descriptions of general processes for decomposition of large systems into subsystems are given by Mesarovick, et al. in [32] for certain types of hierarchical systems. A hierarchical system can be decomposed into a tree or "family" of subsystems in which the "children" subsystems at a given level are independent of their "siblings" at the same level, and the cou-

pling between these sibling subsystems is through the “parent” subsystem. The parent subsystem controls its children subsystems using a set of *coordination variables*. Several methods of forming coordinable subproblems are described in [32]. These methods ensure that when a solution to the decomposed problem is found, it is also a solution to the original problem. The hierarchical decomposition methods of Mesarovick become feasible when the design variables of the problem can be grouped into sets that are only coupled through a few constraint equations (i.e., when there is constraint coupling). One common method for coordination when there is constraint coupling is that of *goal coordination*. In goal coordination, the constraints in a subproblem that have no explicit dependence on the sibling subsystems are treated explicitly, while the coupling constraint equations are multiplied by their Lagrange multipliers and are added to the objective functions, forming a partial Lagrangian function. The parent subsystem, which is the dual of the original problem, determines the optimum value of these Lagrange multipliers. Because goal coordination is essentially a primal-dual optimization method, the objective functions of the child subsystems must be convex, or must be “convexified” [7] for the method to yield satisfactory solutions [17]. This approach has been utilized for plastic design problems by Kaneko and Ha in [23].

The most common decomposition approach for structures treats a structure in a hierarchical way with increasing levels of detail at each lower level in the hierarchy. For example, a discrete region – commonly referred to as a *substructure* – of a stiffened structure may be represented at an upper level in the hierarchy by smeared stiffnesses, while the region is modeled with discrete geometry at a lower level. The decomposition of a structure into substructures

to define optimization regions is described in this section. The decomposition of a structure into substructures for analysis and optimization purposes is described in a subsequent chapter.

Two early approaches that utilized two levels of representation for substructures and that performed searches in the lower-level design space are described by Giles in [16] for wing substructures, and by Kirsch, et al. in [25] for more general substructures. Schmit and Ramanathan [49] describe a two-level approach in which optimization is used within both levels. The upper-level subproblem utilizes weight as the objective function, and area (for trusses) or smeared orthotropic layer thicknesses (for stiffened skins) as design variables. The upper-level constraints are the side constraints (lower and upper limits) on the design variables, displacement constraints, system buckling constraints, and stress constraints. In the upper-level stress constraints, the compressive allowable stress is the maximum of a fixed lower stress limit and of the local buckling and crippling allowables that are determined from the lower-level design of the previous iteration. The lower-level subproblems utilize the discrete geometry of substructures (such as stiffener blade height) as design variables, and minimize the difference of the structural stiffness determined using the lower-level design variables from the stiffness determined using the upper-level design variables. The constraints at this level are discrete geometry side constraints, stress constraints, and local buckling and crippling constraints. At convergence of the two-level approach, the stiffness of the structure is the same whether it is determined by the upper-level or the lower-level design variables. A generalization of this approach that allows for laminated composite structures is described by Schmit and Merhinfar in [47]. In these three two-level

approaches there is no coordination or direct coupling between the lower-level designs of substructures, and the resulting designs are not necessarily optimal.

The linear decomposition method of Sobieski [58] is a widely investigated hierarchical scheme applied to structural optimization that may approach a locally optimal solution. (Although Kirsch [26] and Kirsch and Moses [24] developed rigorous, substructure-based approaches for hierarchical decomposition of a structural optimization problem prior to the work of Sobieski, these methods are not finite-element based, and have only been applied to simple problems.) In the linear decomposition method, the cumulative constraint violation for the lower-level constraints of Schmit and Ramanathan is minimized in each lower-level subproblem. The only explicit constraints in a lower-level subproblem are equality constraints that ensure that the stiffness determined using the lower-level design variables equals the structural stiffness determined using the upper-level design variables. An optimum sensitivity analysis with respect to the upper-level design variables, as described in references [4] and [55], is performed for these lower-level, cumulative-constraint, objective functions. At the upper level, the problem formulation is as described in Schmit and Ramanathan except that the upper-level stress constraints are replaced by linear approximations to the optimal cumulative constraints using the optimization results and the optimal sensitivity analyses of the lower level. This method is demonstrated in [56], generalized for multiple levels in [57], and applied to a large transport aircraft sizing problem in [67]. The method is combined with approximation methods for the constraints in [3], and two more recent variations of the method are described in [61].

Since the linear decomposition method is essentially the decomposition

of a problem using coupling variables (the upper-level design variables enter into the lower-level subproblems as parameters), other solution methods that allow coupling variables can be used to form alternate solution strategies. The penalty approach of Haftka [21], and the method of Thareja and Haftka [64] are two such examples, which interestingly enough can be efficiently solved at a single level. More recently, nonhierarchical decomposition methods have been developed to simplify the multidisciplinary optimization process. These methods utilize the notion of a global sensitivity analysis of interacting subsystems [60] to determine the total sensitivity of interdependent subsystems to changes in problem parameters, and concurrent subspace methods along with an overall coordinating problem for optimization (see references [8], [54], and [59]). Global sensitivity analysis has also been shown to be useful for hierarchical structural analysis by Padula and Polignone in [38].

1.2 Scope of Present Study

In the present dissertation, efficient methods for solving structural optimization problems are developed by approaching an optimization problem using a theoretical framework that is more general than the nonlinear programming theory that forms the basis of the structural optimization methods described in the previous section. The approach of using a different, and perhaps unconventional, theoretical framework to take a fresh look at an old problem is common in mathematics. Utilizing a generalized theoretical framework in studying structural optimization allows for a new perspective of the existing solution methods and suggests new approaches that are more efficient than the existing approaches.

Equilibrium programming (EP), or non-cooperative game theory, is the generalized theoretical framework utilized in the present dissertation. A more detailed overview of EP is presented in the next chapter, but in brief, equilibrium programming is a theory that describes the behavior of multiple, interacting systems that can each be described as NLP problems. An important advantage of utilizing an equilibrium programming framework for developing structural optimization methods over ad-hoc approaches is that a well-developed theory concerning existence and optimality of solutions is available in the literature. Thus, the development of the equilibrium programming structural design formulations in the present dissertation was guided by, and benefitted from, the theoretical structure provided by EP.

Additional guidance for the development of computationally efficient, equilibrium programming structural design formulations is obtained by studying the reasons for the success of the commonly used design methods. As indicated in the previous section, the commonly used methods for solving large-scale structural optimization problems include approximations for the design constraints, various methods for reducing the size of the problem, and decomposition of the problem into multiple subproblems that can be solved in parallel. These methods have been very successful in improving the computational efficiency of finite-element-based structural analysis and optimal structural design. A study of the computational advantages provided by these methods can suggest steps to further improve computational efficiency.

Using approximation concepts, such as simple, approximate equations to describe the design constraint functions, the number of expensive finite element analyses required in the NLP solution is reduced significantly. But a

primary reason for the improved efficiency is that the number of sensitivity analyses, which supply sensitivity derivatives for the gradient-based search methods commonly used, is dramatically decreased. It is often stated that the sensitivity derivatives obtained by the semi-analytic direct sensitivity analysis method (initially developed in [14]) are inexpensive because the factors for the stiffness matrix in the displacement sensitivity equation are available from the structural analysis. However, the right-hand side of the displacement sensitivity equation has as many columns as there are design variables. As a result, the computation of the right-hand side for the semi-analytic sensitivity equation, the back substitution to determine the matrix of displacement sensitivity derivatives, and the use of this matrix in chain rule or finite difference calculations to determine stress sensitivity derivatives can be very expensive. In addition, the computational effort for computing these sensitivity derivatives increases in almost direct proportion to the number of load cases (however, as the number of load cases increases, the adjoint sensitivity method may become more computationally efficient than the direct method). The dominance of sensitivity analysis cost over the structural-response analysis cost is alluded to in [3], and it is explicitly demonstrated for the larger example problems described subsequently. Thus, as problems become more complex, a fundamental need is to reduce the number of sensitivity derivative calculations.

An understanding of the primary advantages of decomposition of an optimization problem can also guide development of methods having increased computational efficiencies. A fundamental advantage of the linear decomposition method is that when a number of lower-level subproblems are to be solved, they can be solved in parallel. Another advantage, which is not often

highlighted, is that the displacement sensitivity derivatives calculated are sensitivity derivatives with respect to a small set of upper-level design variables. In other words, these derivatives are with respect to a relatively small number of substructure stiffnesses, not with respect to the more numerous detailed geometry parameters. Thus, the number of displacement sensitivity derivatives required can be greatly reduced using the linear decomposition method. Even with this advantage, it was determined for the example problem in [67] that 36% of the total computer time for problem solution was utilized in the finite-element-based analysis and sensitivity derivative calculations that involved only five of the 1300 design variables! Thus, although the new optimization methods developed in the present dissertation are derived as equilibrium programming formulations, one of the fundamental approaches for the development of these methods is to find ways to reduce the number of sensitivity analyses required, and to reduce the size of the sensitivity analysis problems.

In subsequent chapters, equilibrium programming theory is summarized, four equilibrium programming structural design formulations are developed, and numerical results for two of these formulations are presented. Linear structural analysis is assumed in the derivations and test problems, and the constraints considered are side constraints, displacement constraints, stress and local buckling constraints. Specifically, some of the fundamental properties of equilibrium programming are summarized in chapter 2. In chapter 3, the basic equations governing finite-element-based structural analysis and optimization are described, and two commonly used design methods, fully stressed design and constraint approximations, are developed as EP design formulations. Two new EP design formulations are also derived in the chapter. The first new EP

design formulation utilizes approximate, updated sensitivity derivatives that replace the constraint sensitivity derivatives determined by a traditional sensitivity analysis at a small fraction of the cost. The second new EP design formulation is a substructure-based decomposition method in which the sensitivity analysis subproblems are greatly reduced in size, and can be solved in parallel. The method of derivation of this formulation ensures the existence and optimality of the solution of the decomposed problem. In chapter 4, some specific information regarding the computer implementation of the EP design formulations is given, and results of using the two new formulations on test problems are obtained and compared with a commonly used method. The test problems for the EP design formulations range from a simple ten-bar truss to a high-speed civil transport having 348 design variables. The overall results are summarized, and suggestions for future work are given in chapter 5.

Chapter 2

Equilibrium Programming Background

In this chapter, the development of equilibrium programming and its previous uses in engineering design are reviewed. The mathematical statement of an equilibrium programming problem, the necessary condition relations, and a solution existence theorem are stated. Some properties of an equilibrium programming solution and some solution methods are also given.

2.1 Equilibrium Programming Background

Equilibrium programming (EP), or non-cooperative game theory, was developed in an operations research setting. The first proof of the existence of the solution to an equilibrium programming problem, called an *equilibrium point*, is due to Nash [34]. In Nash's treatment, the subproblems assume the roles of players trying to maximize their individual pay-off functions in a game, and the domain for each subproblem is a fixed set of strategies that each player can utilize. The formulations of the subproblems are generalized in Debreu [10] so that the domains of each subproblem are functions of the design variables of the other subproblems. Debreu proves the existence of the equilibrium point, subject to certain restrictions, assuming that the feasible domains are nonempty, closed, and bounded regions. Zangwill and Garcia [68] generalize the theorem

of Debreu, and prove existence of equilibrium points under weakened feasibility assumptions.

Although there have been applications of equilibrium programming to economics, game theory, and network theory ([15], pp. 112 - 196), there have been few applications of equilibrium programming in engineering design. Its use in multicriteria engineering design is described by Vincent [65]. In multicriteria design, tradeoffs between possibly conflicting criteria are required so that the design is acceptable to all the designers involved in the design process. Drawbacks in using a simplistic equilibrium programming approach to multicriteria design (i.e., allowing each designer to independently optimize his own design objective function) are nonconvergence of the design, or convergence to a design that is inferior to other possible designs for all the designers. Imposition of an overall coordinating management of the design process may be required to obtain satisfactory designs. One coordination method for the game theory approach to multicriteria design is developed in [42] and is applied to illustrative structures-controls multicriteria design problems in [41]. Thus, an equilibrium programming formulation that includes some form of coordination appears necessary for equilibrium programming to be useful in engineering design. In the EP design formulations derived in the present dissertation, the minimum-weight structural optimization problem, which is essentially a NLP problem, is reformulated as an EP problem. Coordination of the EP subproblems so that the solution is optimal can be important.

2.2 Features of Equilibrium Programming Problems

The theory of equilibrium programming provides a framework to analyze multiple, interdependent nonlinear programming problems. Following Zangwill and Garcia [68], equilibrium programming is a generalization of nonlinear programming (NLP) which can be personified as having M decision makers (which may be implemented as search algorithms) that interact in a system. Each decision maker has a NLP subproblem to solve, and an independent set of design variables to control. The mathematical statement of this problem follows. The design variables controlled by decision maker i are denoted as \mathbf{x}_i , the design variables of all M decision makers are denoted as $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_M)$, and all design variables not controlled by decision maker i are denoted as $\bar{\mathbf{x}}_i = (\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \dots, \mathbf{x}_M)$. Decision maker i has an objective function to minimize, $f_i(\mathbf{x}_i, \bar{\mathbf{x}}_i)$, while satisfying a set of constraints. Thus, the mathematical description of equilibrium programming is:

$$\begin{aligned} & \min_{\mathbf{x}_i} f_i(\mathbf{x}_i, \bar{\mathbf{x}}_i) \\ \text{subject to: } & \mathbf{g}_i(\mathbf{x}_i, \bar{\mathbf{x}}_i) \leq \mathbf{0} \\ & \mathbf{h}_i(\mathbf{x}_i, \bar{\mathbf{x}}_i) = \mathbf{0} \end{aligned} \tag{2.1}$$

for the $i = 1, \dots, M$ interacting NLP subproblems. The variables following the comma in any of the functions in statement (2.1) are treated as fixed parameters in that subproblem. Thus, in the NLP problem of decision maker i , the design variables from other decision makers $\bar{\mathbf{x}}_i$ enter as parameters that account for the coupling of the subproblems.

Any value \mathbf{x} that is a solution to all the NLP subproblems represented by statement (2.1) is called an equilibrium point. There may be numerous equi-

librium points of an equilibrium programming problem. In a manner similar to NLP, first-order necessary conditions are satisfied at an equilibrium point subject to a constraint qualification. Thus, at an equilibrium point, there exist Lagrange multipliers (λ_i, μ_i) such that the following conditions are satisfied:

$$\begin{aligned}
\frac{\partial f_i(\mathbf{x}_i, \bar{\mathbf{x}}_i)}{\partial \mathbf{x}_i} + (\lambda_i)^T \frac{\partial \mathbf{g}_i(\mathbf{x}_i, \bar{\mathbf{x}}_i)}{\partial \mathbf{x}_i} \\
+ (\mu_i)^T \frac{\partial \mathbf{h}_i(\mathbf{x}_i, \bar{\mathbf{x}}_i)}{\partial \mathbf{x}_i} &= \mathbf{0} \\
\mathbf{g}_i(\mathbf{x}_i, \bar{\mathbf{x}}_i) &\leq \mathbf{0} \\
\mathbf{h}_i(\mathbf{x}_i, \bar{\mathbf{x}}_i) &= \mathbf{0} \\
\lambda_i &\geq \mathbf{0} \\
(\lambda_i)^T \mathbf{g}_i(\mathbf{x}_i, \bar{\mathbf{x}}_i) &= 0
\end{aligned} \tag{2.2}$$

for $i = 1, \dots, M$. The conditions governing the existence of solutions $(\mathbf{x}, \lambda, \mu)$ that satisfy the necessary condition relations (2.2) and solve the equilibrium programming problem (2.1) are discussed in the next section.

2.3 Existence of an Equilibrium Point

The satisfaction of a constraint qualification is required for both the existence of a solution to the first-order conditions represented by statement (2.2) and for the existence of an equilibrium point. One form for the constraint qualification is given in [68]. The constraint qualification is satisfied if, for all \mathbf{x} feasible to the NLP subproblems represented by statement (2.1), and for every subproblem i : 1) the vectors $\partial h_{i,j}(\mathbf{x}_i, \bar{\mathbf{x}}_i)/\partial \mathbf{x}_i$ for all components j of \mathbf{h}_i are linearly independent, and 2) there is at least one solution \mathbf{z}_i to the relations:

$$\frac{\partial \mathbf{g}_i^*(\mathbf{x}_i, \bar{\mathbf{x}}_i)}{\partial \mathbf{x}_i} \mathbf{z}_i < \mathbf{0} \tag{2.3}$$

$$\frac{\partial \mathbf{h}_i(\mathbf{x}_i, \bar{\mathbf{x}}_i)}{\partial \mathbf{x}_i} \mathbf{z}_i = \mathbf{0}$$

where \mathbf{g}_i^* is the vector of inequality constraints in subproblem i that are active at \mathbf{x} . With regard to the inequality constraints, this constraint qualification essentially states that it is always possible to move into the interior of the feasible region from a point on the boundary of that region. However, because constraint qualification relations (2.3) must be satisfied individually by each EP subproblem, these requirements on the constraint functions in EP are more restrictive than in NLP. For example, if the constraints from two EP subproblems are given by $\mathbf{g}_1^*(\cdot, \mathbf{x}_2)$ and $\mathbf{g}_2^*(\mathbf{x}_2, \cdot)$, then relation (2.3) cannot be satisfied for subproblem $i = 1$ because $(\partial \mathbf{g}_1^* / \partial \mathbf{x}_1) \mathbf{z}_1 \equiv \mathbf{0}$. However, relation (2.3) may be satisfied for this example when the constraints and design variables of the subproblems are combined within a single NLP problem (i.e., $(\partial \mathbf{g}^* / \partial \mathbf{x}) \mathbf{z} < \mathbf{0}$ where $\mathbf{g}^* = (\mathbf{g}_1^*, \mathbf{g}_2^*)$ and $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$).

A very general theorem for existence of an equilibrium point that is the solution of problem (2.1) is given in [68]. In this theorem, continuity, but not differentiability, of the objective and constraint functions is required. Other conditions for existence are: 1) the functions satisfy constraint qualification relations (2.3) (actually only a weakened form of the constraint qualification is required); 2) the feasible region is bounded with at least one feasible point \mathbf{x}' for which $\mathbf{g}_i(\mathbf{x}'_i, \bar{\mathbf{x}}_i) < \mathbf{0}$ and $\mathbf{h}_i(\mathbf{x}'_i, \bar{\mathbf{x}}_i) = \mathbf{0}$ for every feasible point \mathbf{x} ; 3) the functions $f_i(\mathbf{x}_i, \bar{\mathbf{x}}_i)$ and $g_{i,j}(\mathbf{x}_i, \bar{\mathbf{x}}_i)$ are convex in \mathbf{x}_i ; and 4) the functions $h_{i,j}(\mathbf{x}_i, \bar{\mathbf{x}}_i)$ are linear in \mathbf{x}_i and, for a given i , have linearly independent gradients. The convexity and linearity restrictions on (f_i, \mathbf{g}_i) and \mathbf{h}_i , respectively, may be relaxed and a solution to the necessary condition relations (2.2) will still exist. However, the solution may not be an equilibrium point. Other versions of the EP

existence theorem are reported in [68]. The sufficient conditions for existence of an equilibrium point are more restrictive than those required for existence of an optimal NLP solution. In addition to the more restrictive constraint qualification, a NLP solution exists with either condition 2) given above, or with condition 2) replaced by the coercive assumption defined by $f(\mathbf{x}) \rightarrow +\infty$ as $|\mathbf{x}| \rightarrow \infty$ ([69], p. 363). Some interesting equilibrium point properties are described in the next section.

2.4 Some Equilibrium Point Properties

Although the differences between an equilibrium point and an optimal point (i.e., the solution of a NLP problem) may appear slight, they are important. The following equilibrium point properties, summarized from [15], illustrate the differences. Several examples of these differences can be found in the reference.

An equilibrium point is, in general, different from an optimal point, even if the same constraints are satisfied and each EP subproblem has the same objective function. This difference can occur because the coupling of the constraint derivatives in the respective necessary condition relations is generally weaker for an EP formulation than for a NLP formulation. An example illustrating the differences follows. Suppose that there are two EP subproblems defined by the statements

$$\begin{aligned} & \min_{x_1} 2x_1 - x_2 \\ \text{subject to: } & 0 \leq x_1 \leq 1 \\ & -x_1 + x_2 \leq 0 \end{aligned} \tag{2.4}$$

and

$$\begin{aligned}
 & \min_{x_2} 2x_1 - x_2 \\
 & \text{subject to: } 0 \leq x_2 \leq 1 \\
 & \quad -x_1 + x_2 \leq 0
 \end{aligned} \tag{2.5}$$

These subproblems have the same objective function and a common constraint. All the points on the line segment connecting (0,0) and (1,1) are equilibrium points. If any point on this line segment is attained during the solution process, the stability property of an equilibrium point ([15], p. 84) will prevent any movement from this point, even though there are neighboring solution points that would decrease the objective functions of both subproblems. Suppose that the constraint relations of statements (2.4) and (2.5) were combined into a single NLP using their common objective function, and with design variables (x_1, x_2) . Then the only optimal point for this NLP problem would be (0,0). EP formulations for modeling a system enable the M subproblems to have M different, and possibly conflicting, objective functions. Thus, in general, the equilibrium points of an EP formulation of a system will differ from the optimal points of an NLP formulation of the system that uses the same constraint relations but has only one objective function.

Additional constraints can affect EP solutions differently than NLP solutions. In NLP, additional constraints generally increase the value of the objective function. However, in EP it is possible for additional constraints to force a coordination or cooperation of the subproblems that reduces the objective functions of all the EP subproblems. In the previous example, if the equilibrium point (0.75, 0.75) were found during the solution process, the

objective functions for both subproblems would be 0.75. If the additional constraint $x_2 < 0.5$ were added to subproblem (2.5), the objective functions at the new equilibrium would be reduced to a value no larger than 0.5. The stability property of an equilibrium point mentioned previously states that a solution to an EP problem does not change for a perturbation of the design variables of a single subproblem from equilibrium values. Under certain restrictive assumptions, the equilibrium point is unique, and therefore the stability of the equilibrium point is global [30]. If the equilibrium point is globally stable, the solution methodologies discussed subsequently are more likely to converge to the equilibrium point.

2.5 Solution Methodologies

Because the existence of an equilibrium point is independent of any particular solution method, an equilibrium point can be obtained in several ways. The most straightforward method is to solve all the subproblems sequentially in some predetermined cyclical order. When the solutions to all the subproblems do not change from the previous cycle, an equilibrium point has been reached. Although this method is used in the present dissertation, it may not converge, as demonstrated in [65]. Two approaches that improve the convergence characteristics of a sequential solution method are: 1) approaches that modify the objective functions so that they are strictly convex (such as the proximal point algorithm [44], or the penalty approach of [62]); and 2) move-limit-control methods. Incomplete convergence of intermediate subproblem solutions may also be used to improve computational efficiency. Another variation of the sequential solution method is to solve the subproblems in a sequence

that is determined by information generated during the solution process. Some subproblems may even be omitted from a particular cycle. However, all the subproblems would need to be present in the last cycle to ensure convergence to an equilibrium point.

In another solution method described in [15] (p. 97), the equilibrium point is found by converting the necessary condition relations (2.2) for each of the subproblems into a so-called Kuhn-Tucker equation set. The set of equations can then be solved using any nonlinear equation method. Homotopy methods are described in [15] as one method for solving these equations.

In yet another solution method, applicable if parallel computation is available, the M subproblems can be executed on M processors in an asynchronous manner. Since the other subproblems communicate with subproblem i only through the parameters $\bar{\mathbf{x}}_i$, the processors are effectively decoupled. Any update of \mathbf{x}_i during the solution process on processor i can be made immediately available to the other processors. This method may only be applicable if the functions in the problem have favorable properties [30], and may require additional controls on the solution process to ensure convergence to an equilibrium point. For example, relaxation techniques are used as a control method to improve convergence properties in [6].

Chapter 3

Development of Equilibrium Programming Structural-Design Formulations

In this chapter, the mathematical definition of a NLP-based, optimal structural sizing problem is stated as a point of departure for the development of equilibrium programming structural-design formulations. Only a single load case is used in the following development; extension to multiple load cases is straightforward. Four EP structural-design formulations are then developed. The formal definition of the EP design variables, \mathbf{x}_i , will be given for each formulation. This formal definition may include both true design variables that can only be determined during the solution of the minimization problems, and behavior variables that can be determined from an analysis within the subproblem after the minimum is found.

The first two EP design formulations developed herein were initially described by the author in [51]. Their primary purpose is to further acquaint the reader with equilibrium programming concepts, and to show that two commonly utilized structural-sizing methods are, in actuality, EP formulations. Thus, EP formulations are currently being used to improve the computational efficiency of structural sizing. In these formulations, the EP subproblems consist of a structural-sizing subproblem (or subproblems for the first formulation),

and a structural-response subproblem for each load case. The first equilibrium programming structural-design formulation developed herein considers only stress and local buckling constraints, and side constraints. This formulation is shown to be equivalent to the method of fully stressed design for rod and membrane elements. The second formulation to be described provides for optimal designs with displacement, stress and local buckling constraints, and side constraints. This formulation is shown to be equivalent to the NLP-based approach to optimal structural sizing using a first-order Taylor approximation of the structural response for a rapid analysis.

In the third EP design formulation to be presented, an EP subproblem is developed that performs an approximate sensitivity analysis for a structural-response subproblem using the results of a single finite element analysis and the sensitivity derivatives of a previous iteration. Thus, the cost for this approximate sensitivity analysis is negligible. This formulation was first reported by the author in [52]. Similar expressions for approximate sensitivity derivatives were developed using an ad hoc method in [31] for use in trajectory optimization, and the present formulation generalizes these expressions and provides a formal basis for their derivation.

In the fourth EP design formulation developed herein, a novel substructuring technique is utilized to decompose the analysis and sensitivity calculations for structural design. The substructural analysis portion of the formulation described herein was first reported by the author in [53]. In this formulation, the structure is divided into substructures, and each substructure has its structural response and sensitivity derivatives determined by a structural-response subproblem. The structural sizing and the coordination

of the structural-response subproblems are determined by a single structural-sizing subproblem.

3.1 Optimal Structural-Sizing Problem Statement

The mathematical descriptions of finite-element-based, linear structural analysis, and optimal structural design are given in this section to provide a point of reference for the ensuing discussions. A single structural load case is assumed in the descriptions.

A minimum potential energy formulation can be used in a finite-element-based structural analysis to calculate the structural displacements and stresses that are required to evaluate a design. Given the structural arrangement, the sizes for all the structural elements, a discretization of the structure into finite elements, and a set of external forces on the discretized structure, the correct structural displacements are those that minimize the potential energy of the structure. Thus, the structural response is the solution to the unconstrained NLP problem given by

$$\min_{\mathbf{u}} \left(1/2 \mathbf{u}^T \mathbf{K} \mathbf{u} - \mathbf{F}^T \mathbf{u} \right) \quad (3.1)$$

where the first term in (3.1) is the strain energy, and the second term is the work due to external forces. The vector \mathbf{u} is a vector of nodal displacements, and \mathbf{F} is a vector of external nodal forces (see the Nomenclature section for a discussion of notation and a full list of symbols). The domain of \mathbf{u} is the entire space R^n (where n is the number of displacement degrees-of-freedom), but the boundary conditions on \mathbf{u} are assumed to be incorporated in the global stiffness matrix \mathbf{K} . The necessary condition relations for the unconstrained minimization problem

represented by statement (3.1) are simply the linear equations:

$$\mathbf{K}\mathbf{u} = \mathbf{F} \quad (3.2)$$

Once the displacements \mathbf{u} have been determined from either of these two statements, stresses $\boldsymbol{\sigma}$ (or stress resultants \mathbf{N}) within every element of the structure can be calculated from the displacements, the element strain-displacement relations, and the element constitutive relations.

A nonlinear programming method that is used for optimal structural design can be personified as having one decision maker (which may be implemented as a search algorithm) with control of a set of design variables given by a vector \mathbf{v} . The goal of the decision maker is to minimize an objective function $W(\mathbf{v})$ while satisfying a set of constraints. The mathematical description of a NLP problem is:

$$\begin{aligned} & \min_{\mathbf{v}} W(\mathbf{v}) \\ & \text{subject to: } \mathbf{g}(\mathbf{v}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{v}) = \mathbf{0} \end{aligned} \quad (3.3)$$

where $\mathbf{g}(\mathbf{v})$ are inequality constraints – which could include simple bounds on the design variables – and $\mathbf{h}(\mathbf{v})$ are equality constraints. A common choice for the objective function $W(\mathbf{v})$ for structural sizing is the weight of the structure. The structural-sizing design variables \mathbf{v} , referred to herein as *sizing variables*, can be the dimensions of the individual elements that explicitly contribute to weight, such as beam dimensions, skin thicknesses, and stiffener dimensions and spacing; or they can be variables which affect the weight in an indirect manner, such as the orientation of fibers in a composite structure. The constraint functions considered in the present study are side constraints (such as minimum

gauge) on the sizing variables $\mathbf{g}^s(\mathbf{v})$, the local buckling and stress constraints $\mathbf{g}^\sigma(\mathbf{v}, \boldsymbol{\sigma}(\mathbf{v}))$, and the displacement constraints $\mathbf{g}^u(\mathbf{u}(\mathbf{v}))$. The displacement constraint functions $\mathbf{g}^u(\mathbf{u}(\mathbf{v}))$ are assumed to have no direct dependence on the sizing variables \mathbf{v} , but the functional form $\mathbf{u}(\mathbf{v})$ indicates an indirect dependence on \mathbf{v} through the structural analysis. The stresses $\boldsymbol{\sigma}(\mathbf{v})$, which are shown to depend on \mathbf{v} in the constraint functions, can have several forms. For example, one can write $\boldsymbol{\sigma}(\mathbf{v}) = \boldsymbol{\sigma}^u(\mathbf{v}, \mathbf{u}(\mathbf{v})) = \boldsymbol{\sigma}^N(\mathbf{v}, \mathbf{N}(\mathbf{v}, \mathbf{u}(\mathbf{v})))$ to show that the functional form for stresses can depend directly on displacements \mathbf{u} , or indirectly on displacements through the stress resultants \mathbf{N} . Very often the side constraints, the displacement constraints, and the stress constraints reduce to simple bounds on the sizing variables, the displacements, and the stresses, respectively. Thus, the NLP approach to structural sizing that utilizes finite-element-based structural analysis can be summarized by:

$$\begin{aligned}
& \min_{\mathbf{v}} W(\mathbf{v}) \\
& \text{subject to: } \mathbf{g}^s(\mathbf{v}) \leq \mathbf{0} \\
& \quad \mathbf{g}^u(\mathbf{u}(\mathbf{v})) \leq \mathbf{0} \\
& \quad \mathbf{g}^\sigma(\mathbf{v}, \boldsymbol{\sigma}^u(\mathbf{v}, \mathbf{u}(\mathbf{v}))) \leq \mathbf{0}
\end{aligned} \tag{3.4}$$

where the displacements $\mathbf{u}(\mathbf{v})$ are found from statement (3.1) or equation (3.2). The necessary conditions for the structural-sizing problem (3.4) are the same as given in (2.2) restricted to a single subproblem.

3.2 Structural-Design Formulation 1

In this section, a simplistic approach to defining an equilibrium programming formulation for structural design is described, its limitations are outlined, and

modifications that overcome the limitations are developed. In this approach, one EP subproblem is a structural-response subproblem that is defined by the minimization problem of statement (3.1). In this subproblem, denoted as subproblem 1, the design variables are the displacements (i.e., $\mathbf{x}_1 \equiv \mathbf{u}$), and the sizing variables \mathbf{v} are treated as fixed parameters. By parallel reasoning, a second subproblem, denoted as subproblem 0, could then be identified as a structural-sizing subproblem represented by statement (3.4) in which the design variables are the sizing variables (i.e., $\mathbf{x}_0 \equiv \mathbf{v}$), and the displacements \mathbf{u} are treated as fixed parameters. The shortcoming of this EP formulation is that an equilibrium point may not exist because constraint qualification relations (2.3) cannot be satisfied for some constraints. For example, maximum stress constraints for rod elements having cross-sectional areas \mathbf{A} as design variables (i.e., $\mathbf{x}_0 \equiv \mathbf{v} \equiv \mathbf{A}$) are given by $\mathbf{g}_0^\sigma(\mathbf{x}) \equiv \boldsymbol{\sigma}(\mathbf{u}) - \boldsymbol{\sigma}_{\max} \leq \mathbf{0}$. These constraints depend only on the displacements (\mathbf{x}_1), and not on the areas (\mathbf{x}_0); so there is no \mathbf{z}_0 that will satisfy the inequality in relation (2.3). Thus, a more sophisticated formulation is required.

A formulation that uses an alternate form of the stress constraints may enable satisfaction of the constraint qualification relations. Since functions for calculating the stress and buckling constraints can be constructed by using both the sizing variables and the stress resultants of a structure, a change of variable is made to utilize stress resultants in the constraint functions. Because the stress and buckling constraint functions depend explicitly on the sizing variables with this change of variable, the satisfaction of the constraint qualification is much more likely, but solution existence is still not guaranteed as will be shown in a subsequent example.

The stress resultants throughout the discretized structure are computed within the structural-response subproblems, and are symbolically represented by the equation

$$\mathbf{N} = \mathbf{N}(\mathbf{v}, \mathbf{u}) \quad (3.5)$$

Thus, in this structural-design formulation, the structural-response subproblem (subproblem 1) consists of a solution of the unconstrained minimization given by statement (3.1) (or its necessary conditions equation(3.2)) for \mathbf{u} followed by calculation of stress resultants \mathbf{N} by equation (3.5). The design variables of the structural-response subproblem are defined as $\mathbf{x}_1 \equiv (\mathbf{u}, \mathbf{N})$. The stress resultants \mathbf{N} are used instead of displacements in formulating the stress and buckling constraints of the structural-sizing subproblem (subproblem 0). Thus, the structural-sizing subproblem in this equilibrium programming formulation is:

$$\begin{aligned} & \min_{\mathbf{x}_0 \equiv \mathbf{v}} W(\mathbf{v}) \\ & \text{subject to: } \mathbf{v}_{\min} - \mathbf{v} \leq \mathbf{0} \\ & \mathbf{g}_0^\sigma(\mathbf{v}, \sigma^{\mathbf{N}}(\mathbf{v}, \mathbf{N})) \leq \mathbf{0} \end{aligned} \quad (3.6)$$

where the side constraints are shown as simple minimum gauge constraints in this subproblem.

The method of fully stressed design [28] can be derived from this formulation if: 1) only one-dimensional rod and two-dimensional membrane finite elements are used; 2) one sizing variable is associated with each finite element having a stress constraint; 3) the stress constraints limit the maximum stress magnitude or von Mises stress; and 4) there are no buckling constraints. The structural-sizing subproblem can then be decomposed into a set of independent

elemental problems, one for each sizing variable and constrained element combination. The solution of these elemental problems is simple since the value of the sizing variable that makes a constraint active can be found analytically for each load case. For example, the elemental problem for the cross-sectional area of one-dimensional rod element j is solved by:

$$A_j = \max(A_{j,\min}, |N_j|/\sigma_{j,\max}) \quad (3.7)$$

where $\max(\quad)$ chooses the maximum of its arguments, the structural-sizing design variable $x_{0,j}$ is defined to be the sizing variable A_j , and N_j is the axial force in the element j . Note that the quantities N_j are elements of the stress resultant vector \mathbf{N} , and are also elements of \mathbf{x}_1 and $\bar{\mathbf{x}}_0$. A solution method that alternates between solving the elemental sizing problems, and the structural-response subproblem leads to fully stressed design if it converges.

As stated previously, the change of variables that recasts the stress and buckling constraints in terms of sizing variables and stress resultants makes the existence of an equilibrium point more likely, but not guaranteed. A simple example makes this statement clear. Assume a rod is fixed between two rigid walls and its temperature is increased; the design problem is to size the rod cross-sectional area A to minimize the weight and to satisfy a maximum stress constraint. The temperature change induces a strain which can be expressed as an equivalent external load that is a function of the stiffness. Thus, the equivalent external load (which is also the rod stress resultant N) is calculated by the equality constraint $h_1 \equiv N - k \Delta T A = 0$ in the structural-response subproblem where the sign of N is defined to be positive for compression, k is a constant of proportionality, and ΔT is the temperature rise. The minimum

gauge and the stress constraints, which are calculated in the structural-sizing subproblem, are given by $g_{0,1} \equiv A_{\min} - A \leq 0$ and $g_{0,2} \equiv N - A\sigma_{\max} \leq 0$, respectively. These constraints are shown in figure 3.1. Since a minimum weight design corresponds to a minimum A that satisfies the constraints, two conditions can be identified in the figure. For a relatively low value of ΔT_1 , the fully stressed design approach given here will converge to \mathbf{x}^* shown in the figure. If ΔT_2 is above a limiting value, the constraints allow for no feasible region in the design space, a prerequisite for solution existence, and the fully stressed design algorithm would diverge. This example is severe because even a more sophisticated design method would fail for a large enough ΔT because there would be no feasible region in design space.

Because structural design formulation 1 is a form of fully stressed design, it shares the advantages and disadvantages of fully stressed design. The primary advantage is the simple nature of a structural-sizing subproblem that requires no derivatives and is easily decomposed into a set of independent elemental problems. The disadvantages are: 1) there is no mechanism to ensure satisfaction of the necessary conditions of the optimal structural-sizing problem (3.4) so that the resulting equilibrium point may not be an optimal point; and 2) constraints, such as displacement constraints, that have no explicit dependence on the sizing variables are not considered. An EP formulation is desired which satisfies the optimality necessary conditions at an equilibrium point, and can satisfy displacement constraints.

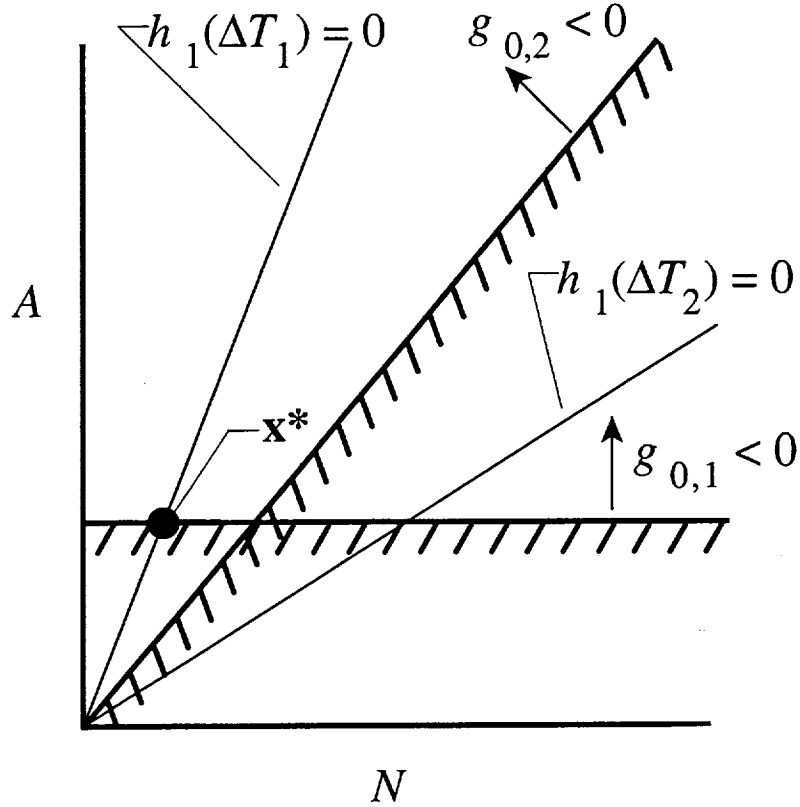


Figure 3.1: Design space for a rod constrained between two rigid walls undergoing a temperature change.

3.3 Structural-Design Formulation 2

If the EP subproblems are identified using the simplistic approach described at the beginning of the previous section, the constraint qualification relations (2.3) cannot be satisfied for any displacement constraint because the displacements would be fixed parameters within the structural-sizing subproblem. In addition, the constraint qualification relations may not be satisfied for certain stress and buckling constraints. Substituting approximate models, that depend explicitly on the sizing variables, for these displacements within the structural-sizing subproblem can overcome these difficulties.

In EP structural-design formulation 2, the displacements \mathbf{u} , which are

parameters independent of the sizing variables in structural-sizing constraint functions $\mathbf{g}(\mathbf{v}, \mathbf{u})$, are replaced with a first-order Taylor series approximation given by

$$\mathbf{u}^A(\mathbf{v}, \mathbf{x}_1) \equiv \frac{\partial \mathbf{u}}{\partial \mathbf{v}} (\mathbf{v} - \mathbf{v}_1) + \mathbf{u} \quad (3.8)$$

In equation (3.8), the matrix $\partial \mathbf{u} / \partial \mathbf{v}$ may be viewed as a matrix of optimal sensitivity derivatives with respect to parameters \mathbf{v} [55] of the displacements determined in the structural-response subproblem (subproblem 1). The value \mathbf{v}_1 is the value of \mathbf{v} utilized in subproblem 1 when the sensitivity derivatives are calculated. All the design variables for the structural-response subproblem are utilized in equation (3.8) since $\mathbf{x}_1 \equiv (\mathbf{u}, \partial \mathbf{u} / \partial \mathbf{v}, \mathbf{v}_1)$. This approximation satisfies the following two properties. First, the approximation depends explicitly on \mathbf{v} so that constraint qualification relations (2.3) necessary for equilibrium point existence can be satisfied. Second, the approximation satisfies the conditions: $\mathbf{u}^A = \mathbf{u}$ and $\partial \mathbf{u}^A / \partial \mathbf{v} = \partial \mathbf{u} / \partial \mathbf{v}$ at the equilibrium point $\mathbf{x} = \mathbf{x}^*$ where $\mathbf{v} = \mathbf{v}_1 = \mathbf{v}^*$. This second set of conditions ensures the optimality of the design given by the equilibrium point because the EP necessary conditions are then the same as the NLP necessary conditions for problem (3.4).

Using the definition of equation (3.8), the structural-sizing subproblem is given by the following statement:

$$\begin{aligned} & \min_{\mathbf{x}_0 \equiv \mathbf{v}} W(\mathbf{v}) \\ & \text{subject to: } \mathbf{v}^l \leq \mathbf{v} \leq \mathbf{v}^u \\ & \mathbf{g}^s(\mathbf{v}) \leq \mathbf{0} \\ & \mathbf{g}^u(\mathbf{u}^A(\mathbf{v}, \mathbf{x}_1)) \leq \mathbf{0} \\ & \mathbf{g}^\sigma(\mathbf{v}, \sigma^u(\mathbf{v}, \mathbf{u}^A(\mathbf{v}, \mathbf{x}_1))) \leq \mathbf{0} \end{aligned} \quad (3.9)$$

where move limits \mathbf{v}^l and \mathbf{v}^u that are adjusted during the solution process have been introduced to ensure convergence. The structural-response subproblem is represented by unconstrained minimization problems given by statement (3.1) (or the necessary conditions given by equation (3.2)), and the following equations (which can be formally treated as equality constraints) that determine the behavior variables $\partial \mathbf{u} / \partial \mathbf{v}$ and \mathbf{v}_1 :

$$\begin{aligned} \mathbf{K}(\mathbf{v}) \frac{\partial \mathbf{u}}{\partial \mathbf{v}} + \frac{\partial \mathbf{K}(\mathbf{v}) \mathbf{u}}{\partial \mathbf{v}} &= [\mathbf{0}] \\ \mathbf{v}_1 &= \mathbf{v} \end{aligned} \quad (3.10)$$

The first of the two equations in statement (3.10) is a sensitivity analysis equation using a direct method of sensitivity analysis ([19], p. 264). In this equation, the sizing variables \mathbf{v} are assumed known, the vector \mathbf{u} in the second term is treated as a constant, and contributions from the nodal forces \mathbf{F} are neglected since \mathbf{F} is assumed to be independent of \mathbf{v} . In practice, the second term of this equation is often determined by finite differences. The second equation in (3.10) is a trivial identity which preserves the values of the structural-sizing variables at which the sensitivity analysis is performed so they can be exported to the other subproblems.

EP structural-design formulation 2 is equivalent to a NLP approach to structural design with a first-order Taylor series for an approximate analysis (i.e., the method of [45] except that reciprocal variables are not used) if the equilibrium point is found by the following sequential steps: 1) solve the subproblems represented by statement (3.1) (or necessary conditions represented by statement (3.2)) for \mathbf{u} ; 2) solve equations (3.10) for $\partial \mathbf{u} / \partial \mathbf{v}$ and \mathbf{v}_1 ; 3) use the quantities found in steps 1 and 2 in equation (3.8) to form approximate

models for displacements $\mathbf{u}^A(\mathbf{v}, \mathbf{x}_1)$; 4) utilize the approximate models of step 3 in the structural-sizing subproblem represented by statement (3.9) to solve for \mathbf{v} ; and 5) repeat steps 1 through 4 in a cyclic manner, with an algorithm defined to update the move limits, until the changes in the solutions from consecutive cycles converge. Although first-order Taylor series are used in this development, other approximate models for displacements that depend on first derivatives could be used in place of (3.8) ([19], pp. 211–219).

3.4 Structural-Design Formulation 3

The two EP design formulations described previously represent current practices in optimal structural sizing. To further improve computational efficiency, a formulation is derived herein that computes approximate, updated sensitivity derivatives in an EP subproblem. The cost of computing these approximate derivatives is essentially only the cost of the structural analysis, and the evaluation of the design constraints.

The approximate sensitivity analysis is derived as a correction, or *update*, to previous values for the sensitivity derivatives. The sensitivity-derivative-update subproblem is described for a general scalar function $f(\mathbf{v})$, where \mathbf{v} is a vector quantity, as follows. The values of f and the gradient of f , $\partial f / \partial \mathbf{v}$, are assumed to be known at a previous value of \mathbf{v} . These quantities are denoted as f_{old} , $\partial f_{\text{old}} / \partial \mathbf{v}$, and \mathbf{v}_{old} , respectively. The value of f at the new value, $\mathbf{v} = \mathbf{v}_{\text{new}}$, is also known and an approximation to the gradient of f at $\mathbf{v} = \mathbf{v}_{\text{new}}$ is sought; this approximate gradient is denoted as $\partial f_{\text{new}} / \partial \mathbf{v}$. The difference between $\partial f_{\text{new}} / \partial \mathbf{v}$ and $\partial f_{\text{old}} / \partial \mathbf{v}$ is given by the gradient updating vector \mathbf{a} (also called an update vector, with components that are called updates herein), and

the relation among these quantities is expressed by the equation

$$\frac{\partial f_{\text{new}}}{\partial \mathbf{v}} = \frac{\partial f_{\text{old}}}{\partial \mathbf{v}} + \mathbf{a} \quad (3.11)$$

Using a criterion similar to that utilized in [11] (p. 171) in calculating least-change secant updates of the Jacobian matrix for solving simultaneous equations, the update vector \mathbf{a} is chosen to be the vector that has the smallest (weighted) magnitude, and that also satisfies a second-order Taylor series relation between the quantities at $\mathbf{v} = \mathbf{v}_{\text{new}}$ and $\mathbf{v} = \mathbf{v}_{\text{old}}$. The constrained minimization problem that expresses these conditions is:

$$\begin{aligned} & \min_{a_k} \sum_k (a_k w_k)^2 \\ \text{subject to: } & f_{\text{new}} = f_{\text{old}} + \sum_k \left(\frac{\partial f_{\text{old}}}{\partial v_k} + \frac{1}{2} a_k \right) (v_{\text{new},k} - v_{\text{old},k}) \end{aligned} \quad (3.12)$$

where w_k are user-defined weighting parameters. The equality constraint in the minimization problem given by statement (3.12) indicates that the quantities a_k are approximations to $\sum_l \partial^2 f / \partial v_k \partial v_l (v_{\text{new},l} - v_{\text{old},l})$. Thus, the accuracy of the sensitivity derivatives determined by the update vector is expected to degrade as the quantity $|\mathbf{v}_{\text{new}} - \mathbf{v}_{\text{old}}|$ increases in value. Letting $\Delta v_k \equiv v_{\text{new},k} - v_{\text{old},k}$ and $\Delta f \equiv f_{\text{new}} - f_{\text{old}} - \sum_k \partial f_{\text{old}} / \partial v_k \Delta v_k$, the solution to the minimization problem given by statement (3.12) can be found analytically using a Lagrange multiplier method

$$a_k = 2 \frac{\Delta f \Delta v_k}{w_k^2 \sum_l \Delta v_l^2 / w_l^2} \quad (3.13)$$

A simplified version of this expression has been developed by an ad hoc method in [31] with application to the approximation of gradients for trajectory optimization. However, the update equations described herein are derived

independently using a formal methodology that allows for the general inclusion of the weights w_k that are not considered in [31]. For identical weights w_k , the update given by equation (3.13) is twice the Broyden update equation described in [11] (p. 170), and is the same as that derived in [31]. The advantage of including the general weighting terms is described subsequently. In the present study, the function f represents a component of the constraint function vector \mathbf{g} .

3.4.1 Structural-Design Formulation Using Updated Sensitivity Derivatives

The method of approximate, updated sensitivity derivatives is denoted as EP structural-design formulation 3 (or simply formulation 3) herein. This formulation is a modification of design formulation 2 described previously in which update subproblems (3.12) are utilized during the solution process in place of computing sensitivity derivatives such as (3.10). In subproblem 0 (the structural-sizing subproblem), the design variables are $\mathbf{x}_0 = (\mathbf{v}, \mathbf{v}_0, \mathbf{g}_0, \partial \mathbf{g}_0 / \partial \mathbf{v})$. All but the first of which are actually behavior variables in that subproblem. Formally, this subproblem can be represented by the following statement:

$$\begin{aligned}
 & \min_{\mathbf{v}} W(\mathbf{v}) \\
 \text{subject to: } & \frac{\partial \mathbf{g}_1}{\partial \mathbf{v}} (\mathbf{v} - \mathbf{v}_1) + \mathbf{g}_1 \leq \mathbf{0} \\
 & \mathbf{v}^l \leq \mathbf{v} \leq \mathbf{v}^u \\
 & \mathbf{g}^s(\mathbf{v}) \leq \mathbf{0} \\
 & \mathbf{v}_0 = \mathbf{v}_1 \\
 & \mathbf{g}_0 = \mathbf{g}_1
 \end{aligned} \tag{3.14}$$

$$\frac{\partial \mathbf{g}_0}{\partial \mathbf{v}} = \frac{\partial \mathbf{g}_1}{\partial \mathbf{v}}$$

where the general constraints are linearized, and move limits \mathbf{v}^l and \mathbf{v}^u that are adjusted during the solution process have been introduced to ensure convergence. The last three equalities in problem (3.14) are required to define data items that are saved to be utilized in the structural-response subproblem. Subproblem 1 (the structural-response subproblem) has design variables given by $\mathbf{x}_1 = (\mathbf{u}, \mathbf{g}_1, \mathbf{v}_1, \partial \mathbf{g}_1 / \partial \mathbf{v})$, and most of these quantities can be computed after solving for the first member of \mathbf{x}_1 (i.e., \mathbf{u}). The subproblem statement is given as:

$$\begin{aligned} \min_{\mathbf{u}} \quad & \left(1/2 \mathbf{u}^T \mathbf{K}(\mathbf{v}) \mathbf{u} - \mathbf{F}^T \mathbf{u} \right) \\ \text{subject to:} \quad & \mathbf{g}_1 = \mathbf{g}(\mathbf{v}, \mathbf{u}) \\ & \mathbf{v}_1 = \mathbf{v} \\ & SENS(choice) = 0 \end{aligned} \tag{3.15}$$

where the expression $SENS(choice) = 0$ represents the conditions:

$$\begin{aligned} \mathbf{K}(\mathbf{v}) \frac{\partial \mathbf{u}}{\partial \mathbf{v}} &= - \frac{\partial \mathbf{K}(\mathbf{v}) \mathbf{u}}{\partial \mathbf{v}} \\ \frac{\partial g_{1,j}}{\partial v_i} &= \frac{g_j((\mathbf{v} + \Delta v_i \mathbf{e}_i^{\mathbf{v}}), (\mathbf{u} + \partial \mathbf{u} / \partial \mathbf{v} \Delta v_i \mathbf{e}_i^{\mathbf{v}})) - g_{1,j}}{\Delta v_i} \end{aligned} \tag{3.16}$$

when “exact” sensitivity analysis is desired, and

$$\frac{\partial g_{1,j}}{\partial v_i} = \frac{\partial g_{0,j}}{\partial v_i} + 2(v_i - v_{0,i}) \left(\frac{\partial g_{0,j}}{\partial v_i} \right)^2 \frac{g_{1,j} - g_{0,j} - \partial g_{0,j} / \partial \mathbf{v} (\mathbf{v} - \mathbf{v}_0)}{\sum_l (v_l - v_{0,l})^2 (\partial g_{0,j} / \partial v_l)^2} \tag{3.17}$$

when approximate sensitivity analysis is desired. Several implementation features of this formulation should be noted. The potential energy minimization in (3.15) is a formal statement, and may be replaced with the necessary conditions

of equation (3.2). The equality constraints in (3.15) are formal definitions to define quantities needed in the structural-response subproblem. As in the determination of the move limits, an additional algorithm is required to determine whether statement (3.16) or statement (3.17) is utilized for $SENS(choice) = 0$ in statement (3.15) at each stage in the solution process. Because the update minimization subproblems have analytic solutions given by equation (3.13), these solutions are incorporated as constraints in the structural-response subproblem using equation (3.17) instead of specifying them as independent subproblems. Finally, the weights in the update subproblems have been specified in equation (3.17) as the inverse of the gradients, $w_i = \partial g_0 / \partial v_i$. Thus, the updates to the sensitivity derivatives determined from the subproblems given by statement (3.12) minimize the *relative* changes of the sensitivity derivatives from their previous values, and the update to each component of a gradient vector is related to the component magnitude. One advantage of this weighting method is that the sparsity pattern of sensitivity derivatives is maintained since an entry that is zero or small in magnitude remains zero or small in magnitude after updating. This method of weighting was also found to provide better convergence than using equal weights in some preliminary optimization studies. More details concerning the computer implementation of this design formulation are discussed in the next chapter.

3.5 Design Formulation 4

In this section, an EP design formulation is described that utilizes decomposition methods when computing the structural response and the sensitivity derivatives. In this formulation, the structure is divided into substructures, and

each substructure has its structural response described by a structural-response subproblem. The structural sizing and the coordination of the structural-response subproblems are determined by a single structural-sizing subproblem.

Substructure-based design methods have been utilized in [25] and [26], but the analyses in these references were not based on substructure principles. Traditional (i.e., superelement), substructure-based sensitivity methods for structural optimization have been used to form a reduced basis for approximate analysis in [36], have been derived using the adjoint sensitivity method in [1], and have been combined with kinematic constraints to reduce the number of interface degrees of freedom in [37]. The adjoint method of substructure sensitivity analysis of [1] is generalized to multiple levels of substructures in [35]. An alternate method of using substructures for performing structural analysis is developed in [12]. The method of reference [12] is essentially a substructure-based, hierarchical decomposition method in which the response of each substructure is determined by independent subproblems. Coupling constraints that determine the interface forces required to enforce the compatibility of common interface nodes form a coordination problem. Decomposition into substructures that are not restrained from rigid-body motions yields subproblems that are not strictly convex. Nonconvexity causes difficulties, as noted in [17], that can be handled using special techniques for solving both the subproblems for each substructure and the coordination problem. A modification of this alternate substructure analysis method, described in [13], incorporates penalty functions to ensure strict convexity of the subproblem objective functions.

The EP design formulation of the present section is similar to the

method of [12], but strict convexity is not required in this approach. In the present derivation, each design constraint is assumed to be local so that it depends on the sizing variables and the structural response of a single substructure. Thus, stress constraints, local buckling constraints, and constraints that are functions of displacements within single substructures may be considered in this approach. In the next subsection, the decomposition of the structural analysis is discussed utilizing a two-substructure example. Then the derivation is extended to an arbitrary number of substructures, and the substructure-based EP design formulation is developed.

3.5.1 Derivation of the Substructure Analysis Method

The method of determining the structural response is presented for a structure that is decomposed into substructures, each of which is assumed to have a linear elastic response with no zero-strain-energy motions possible except for rigid-body motions. To permit a simple exposition of the method, it is initially derived using two substructures. Four salient features characterize the present method for structural analysis using substructures. Firstly, when the substructure is not fully restrained from rigid-body motions, the structural response is decomposed into the rigid-body motions (referred to as “modes” herein) of the substructure, and displacements orthogonal to the rigid-body motions (i.e., the elastic deformations). Secondly, an augmented stiffness matrix is formed for each of these substructures. These stiffness matrices are symmetric, and can be factored independently and, computationally, in parallel. Thirdly, a structural-response coordination problem determines the forces between the substructures, and the magnitude of the rigid-body modes. The structural-

response coordination problem requires the factored, stiffness matrices of the substructures, and results in a system of linear equations with its order equal to the number of shared degrees-of-freedom between substructures plus the total number of substructure rigid-body modes. Lastly, once the forces between the substructures are determined, the displacements orthogonal to the rigid-body modes can be determined.

The simple wing structure finite element model in figure 3.2 is utilized to demonstrate the derivation of the analysis decomposition method. The model shown in the figure is decomposed into two substructures, the second of which is unrestrained and thus has six rigid-body modes. The linear elastic structural response for the entire wing is determined using a minimum potential energy formulation. This response is the solution to the unconstrained minimization problem given by statement (3.1) having the necessary conditions given by equation (3.2). In this section, the nodal displacements are denoted using the lower case (i.e., \mathbf{u}). When the displacements of a substructure can be decomposed into elastic deformations and rigid-body modes, the elastic deformations are denoted by the upper case (i.e., \mathbf{U}). Thus, the displacements of the entire wing are denoted by the nodal displacement vector \mathbf{u} , and the displacements of the two substructures are given by the vectors \mathbf{u}_1 and \mathbf{u}_2 . The external loading for the entire wing is given by the nodal force vector \mathbf{F} , and the decomposition of the external loading is given by vectors \mathbf{F}_1 and \mathbf{F}_2 for the two substructures. The dimensions of each of these vectors is the number of nodal degrees of freedom of the appropriate substructure. The nodal displacements at the interface of the two substructures must be equal for the two substructures to be compatible. Those compatible displacements at the interfaces of the two substructures

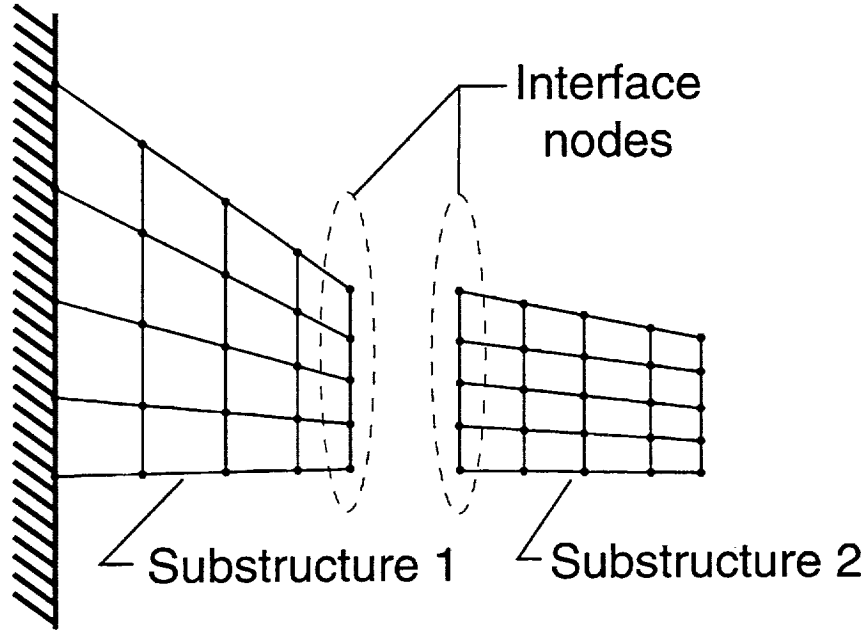


Figure 3.2: Example wing structural finite element model decomposed into two substructures.

are specified in a predetermined order with signed Boolean matrices denoted by \mathbf{B}_1 and \mathbf{B}_2 , where the nonzero entries of \mathbf{B}_1 are equal to $+1$ and the nonzero entries of \mathbf{B}_2 are equal to -1 . The constraints of compatible displacements at the interface between substructures is then expressed by $\mathbf{B}_1 \mathbf{u}_1 + \mathbf{B}_2 \mathbf{u}_2 = \mathbf{0}$. The decomposition of the externally applied nodal forces at the interface nodes is arbitrary as long as the sum of the forces applied to the interface nodes of each substructure is equal to the actual externally applied forces at these nodes.

In a substructural decomposition, a stiffness matrix can be formulated for each substructure. These stiffness matrices are relative to the substructure nodal displacement vectors \mathbf{u}_i , and are denoted by \mathbf{K}_i . Thus, a constrained minimization problem that is equivalent to the problem given by statement

(3.1), and uses the substructure nodal displacement vectors is

$$\begin{aligned} \min_{(\mathbf{u}_1, \mathbf{u}_2)} \quad & \left(\frac{1}{2} \mathbf{u}_1^T \mathbf{K}_1 \mathbf{u}_1 + \frac{1}{2} \mathbf{u}_2^T \mathbf{K}_2 \mathbf{u}_2 \right. \\ & \left. - \mathbf{F}_1^T \mathbf{u}_1 - \mathbf{F}_2^T \mathbf{u}_2 \right) \\ \text{subject to: } & \mathbf{B}_1 \mathbf{u}_1 + \mathbf{B}_2 \mathbf{u}_2 = \mathbf{0} \end{aligned} \quad (3.18)$$

The constrained minimization problem given by statement (3.18) has a separable objective function with coupling constraints, and appears to be a candidate for a hierarchical decomposition into two distinct minimization problems having design variables \mathbf{u}_1 and \mathbf{u}_2 , respectively. The lower-level subproblems for the displacements using such a decomposition are

$$\min_{\mathbf{u}_1} \frac{1}{2} \mathbf{u}_1^T \mathbf{K}_1 \mathbf{u}_1 - \mathbf{F}_1^T \mathbf{u}_1 + \boldsymbol{\lambda}^T \mathbf{B}_1 \mathbf{u}_1 \quad (3.19)$$

and

$$\min_{\mathbf{u}_2} \frac{1}{2} \mathbf{u}_2^T \mathbf{K}_2 \mathbf{u}_2 - \mathbf{F}_2^T \mathbf{u}_2 + \boldsymbol{\lambda}^T \mathbf{B}_2 \mathbf{u}_2 \quad (3.20)$$

where the goal coordination approach described in [32] (p. 240), and summarized in the first chapter of the present dissertation, is utilized with the common coordination inputs $\boldsymbol{\lambda}$. To find the minimum of problem (3.18) using goal coordination, the coordination inputs would need to be determined by a coordination problem that is typically the dual formulation of statement (3.18). However, the minimization problem for (3.20) is insoluble due to the rank deficiency of matrix \mathbf{K}_2 . An EP explanation for the insolubility of subproblem (3.20) can be developed based on the conditions for the existence of an EP solution. One of the sufficient conditions for the existence of an equilibrium point requires a bounded solution space for the subproblems. However \mathbf{u}_2 is not bounded as seen from the following argument. Because substructure 2 has

six rigid-body modes, a rigid-body matrix \mathbf{R}_2 can be formed that consists of the six rigid-body modes as column vectors. This rigid-body matrix satisfies the condition $\mathbf{K}_2\mathbf{R}_2 = [\mathbf{0}]$, thus the vector $\mathbf{R}_2\boldsymbol{\alpha}_2$ added to any displacement \mathbf{u}_2 will not change the substructure potential energy if the vector $\boldsymbol{\alpha}_2$ satisfies the scalar equation $\mathbf{F}_2^T\mathbf{R}_2\boldsymbol{\alpha}_2 = 0$. Since the vector $\boldsymbol{\alpha}_2$ is not otherwise restricted, it (and therefore \mathbf{u}_2) is unbounded. Thus, the sufficient conditions for solution existence are not satisfied. To overcome the difficulties due to rank deficiency of substructures having rigid-body modes, a pseudo inverse of the matrix \mathbf{K}_2 is used in [12] when solving for \mathbf{u}_2 in the necessary conditions for subproblem (3.20). The coordination inputs $\boldsymbol{\lambda}$, and the magnitude of the rigid-body modes are determined in a coordination subproblem. Alternate ways of modifying subproblem (3.20) to make its solution bounded are to add terms to make the objective function strictly convex using the proximal point algorithm as in [7], or using the separable penalty method of [62]. Another penalty method that can convexify the problem is reported in [13].

From the preceding discussion, any decomposition of statement (3.1) should explicitly account for the rigid-body modes of the substructures. One such decomposition method follows. Let $\mathbf{u}_2 = \mathbf{U}_2 + \mathbf{R}_2\boldsymbol{\alpha}_2$ where \mathbf{U}_2 satisfies the supplementary condition $\mathbf{R}_2^T\mathbf{U}_2 = \mathbf{0}$ (i.e., \mathbf{U}_2 is orthogonal to the rigid body modes). The minimization problem given by statement (3.18) then becomes

$$\begin{aligned}
& \min_{(\mathbf{u}_1, \mathbf{U}_2, \boldsymbol{\alpha}_2)} \left(\frac{1}{2}\mathbf{u}_1^T\mathbf{K}_1\mathbf{u}_1 + \frac{1}{2}\mathbf{U}_2^T\mathbf{K}_2\mathbf{U}_2 \right. \\
& \quad \left. - \mathbf{F}_1^T\mathbf{u}_1 - \mathbf{F}_2^T(\mathbf{U}_2 + \mathbf{R}_2\boldsymbol{\alpha}_2) \right) \tag{3.21} \\
& \text{subject to: } \mathbf{B}_1\mathbf{u}_1 + \mathbf{B}_2(\mathbf{U}_2 + \mathbf{R}_2\boldsymbol{\alpha}_2) = \mathbf{0} \\
& \quad \mathbf{R}_2^T\mathbf{U}_2 = \mathbf{0}
\end{aligned}$$

Using a hierarchical decomposition with goal coordination, three lower-level subproblems can be formulated. The first is identical to statement (3.19), the second is

$$\begin{aligned} \min_{\mathbf{U}_2} \left(\frac{1}{2} \mathbf{U}_2^T \mathbf{K}_2 \mathbf{U}_2 - \mathbf{F}_2^T \mathbf{U}_2 + \boldsymbol{\lambda}^T \mathbf{B}_2 \mathbf{U}_2 \right) \\ \text{subject to: } \mathbf{R}_2^T \mathbf{U}_2 = \mathbf{0} \end{aligned} \quad (3.22)$$

and the third is

$$\min_{\boldsymbol{\alpha}_2} -\mathbf{F}_2^T \mathbf{R}_2 \boldsymbol{\alpha}_2 + \boldsymbol{\lambda}^T \mathbf{B}_2 \mathbf{R}_2 \boldsymbol{\alpha}_2 \quad (3.23)$$

The objective function of subproblem (3.22) is strictly convex relative to the subspace $\{\mathbf{U}_2 \mid \mathbf{R}_2^T \mathbf{U}_2 = \mathbf{0}\}$, and thus the subproblem will have a bounded solution. However, subproblem (3.23) will have no finite solution unless $\boldsymbol{\lambda}$ satisfies the restriction $-\mathbf{F}_2^T \mathbf{R}_2 + \boldsymbol{\lambda}^T \mathbf{B}_2 \mathbf{R}_2 = \mathbf{0}^T$. This restriction on $\boldsymbol{\lambda}$ becomes a constraint in the standard goal coordination subproblem that maximizes the Lagrangian function (as a function of $\boldsymbol{\lambda}$) in the dual formulation of statement (3.21). This restriction is also used in the conjugate projected gradient approach to solving the coordination subproblem in [12]. One expression for the coordination subproblem can be found from the following necessary conditions for the structural-response problem given by statement (3.21)

$$\mathbf{K}_1 \mathbf{u}_1 - \mathbf{F}_1 + \mathbf{B}_1^T \boldsymbol{\lambda} = \mathbf{0} \quad (3.24)$$

$$\tilde{\mathbf{K}}_2 \left[\frac{\mathbf{U}_2}{\boldsymbol{\mu}} \right] - \left[\frac{\mathbf{F}_2 - \mathbf{B}_2^T \boldsymbol{\lambda}}{\mathbf{0}} \right] = \mathbf{0} \quad (3.25)$$

$$\mathbf{R}_2^T (\mathbf{F}_2 - \mathbf{B}_2^T \boldsymbol{\lambda}) = \mathbf{0} \quad (3.26)$$

$$\mathbf{B}_1 \mathbf{u}_1 + \mathbf{B}_2 (\mathbf{U}_2 + \mathbf{R}_2 \boldsymbol{\alpha}_2) = \mathbf{0} \quad (3.27)$$

where $\boldsymbol{\lambda}$ are the Lagrange multipliers for the first constraint in statement (3.21), $\boldsymbol{\mu}$ are the Lagrange multipliers for the second constraint in statement (3.21),

and the matrix $\tilde{\mathbf{K}}_2$ is the augmented stiffness matrix given by

$$\tilde{\mathbf{K}}_2 = \left[\begin{array}{c|c} \mathbf{K}_2 & \mathbf{R}_2 \\ \hline \mathbf{R}_2^T & [0] \end{array} \right] \quad (3.28)$$

Because the matrices \mathbf{B}_i in necessary conditions (3.24-3.27) are Boolean, the Lagrange multipliers $\boldsymbol{\lambda}$ are simply the interface forces between the substructures. Using equation (3.25), the condition $\mathbf{R}_2^T \mathbf{K}_2 = [0]$, and the fact that the columns of \mathbf{R}_2 are linearly independent, equation (3.26) implies that the Lagrange multipliers $\boldsymbol{\mu}$ are equal to zero. Note that equation (3.24) is the necessary condition equation for subproblem (3.19), and equation (3.25) is the necessary condition equation for subproblem (3.22).

The structural-response coordination subproblem that determines $\boldsymbol{\lambda}$ and $\boldsymbol{\alpha}_2$ is obtained by substituting the solutions of equations (3.24) and (3.25) into equations (3.26) and (3.27). This substitution yields the equation

$$\mathbf{M} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha}_2 \end{bmatrix} = \left[\begin{array}{c} \mathbf{B}_1 \mathbf{K}_1^{-1} \mathbf{F}_1 + [\mathbf{B}_2 \mid [0]] \tilde{\mathbf{K}}_2^{-1} [\mathbf{F}_2^T \mid \mathbf{0}^T]^T \\ \hline -\mathbf{R}_2^T \mathbf{F}_2 \end{array} \right] \quad (3.29)$$

where the matrix \mathbf{M} is given by

$$\mathbf{M} = \left[\begin{array}{c|c} \mathbf{B}_1 \mathbf{K}_1^{-1} \mathbf{B}_1^T + [\mathbf{B}_2 \mid [0]] \tilde{\mathbf{K}}_2^{-1} [\mathbf{B}_2 \mid [0]]^T & -\mathbf{B}_2 \mathbf{R}_2 \\ \hline -(\mathbf{B}_2 \mathbf{R}_2)^T & [0] \end{array} \right] \quad (3.30)$$

Note that the inverse of $\tilde{\mathbf{K}}_2$ used in equations (3.29) and (3.30) exists because of the inclusion of the constraint that \mathbf{U}_2 is orthogonal to the rigid-body modes. Thus, one approach to the calculation of the structural response (i.e., the nodal displacements) is the following steps: 1) factor the matrices \mathbf{K}_1 and $\tilde{\mathbf{K}}_2$; 2) use these factored matrices to formulate the matrix \mathbf{M} and the right hand side of equation (3.29); 3) solve equation (3.29) for $\boldsymbol{\lambda}$ and $\boldsymbol{\alpha}_2$; and 4) and solve subproblem (3.19) (or equation (3.24)) and subproblem (3.22) (or equation

(3.25)) for \mathbf{u}_1 and \mathbf{U}_2 , respectively. The explicit computation of matrix \mathbf{M} is not necessary in the combined analysis and optimization approach described in the next section.

3.5.2 Derivation of EP Design Formulation 4

In this subsection, the structural-optimization problem is decomposed into subproblems that perform the structural analyses and sensitivity analyses of each substructure and a single subproblem that performs the coordination of the structural analyses as well as the optimization of the design. This formulation is denoted as EP design formulation 4. The decomposition method for structural analysis derived in the previous section is extended to multiple substructures. Then the design formulation is derived starting with a simultaneous analysis and design formulation. The simultaneous analysis and design method was initially investigated in [46]. It is utilized in [22] for linear structural analysis, and in [20] for nonlinear structural analysis. In the present simultaneous analysis and design formulation, in which minimum weight is the design goal, both the structural displacements and the sizing variables are utilized as design variables, and the optimization constraints are the equations governing structural response (treated as equality constraints) as well as the usual inequality constraints that ensure that the design meets the strength, buckling, and other design requirements. Formulation 4 has the following features: 1) the analysis and sensitivity derivatives of each substructure are independent and can be performed in parallel (although the overall optimization procedure is still iterative); 2) the resulting design is optimal; but 3) the structural response may not be compatible between the substructures until the design converges.

Structural Analysis using Multiple Substructures

The substructural analysis method developed previously is generalized to multiple substructures in this subsection. With multiple substructures, the use of precise, although somewhat cumbersome, notation is necessary to avoid confusion. Assume that there are M substructures, and that there are m substructures ($0 < m < M$) unrestrained from rigid-body motions if separated from the rest of the structure. The n_i displacement degrees-of-freedom of substructure i are denoted by $\mathbf{u}_i = (u_{i,1}, \dots, u_{i,n_i})^T$. The substructures unrestrained from rigid-body motions are ordered to be the last m substructures. These substructures have displacements denoted by $\mathbf{u}_i = \mathbf{U}_i + \mathbf{R}_i \boldsymbol{\alpha}_i$ where \mathbf{R}_i is the matrix containing the rigid-body modes of substructure i , and \mathbf{U}_i are displacements orthogonal to these rigid-body modes. The orthogonality relation for \mathbf{U}_i is expressed by $\mathbf{R}_i^T \mathbf{U}_i = \mathbf{0}$. There are n^I independent relations governing the compatibility of the substructures, and the resultant equivalencing of degrees-of-freedom at common interface nodes in the different substructures is represented by the set A of n^I 4-tuples defined by $A = \{ (j, k, p, q) \mid j < k \text{ and } u_{j,p} \equiv u_{k,q} \}$. In the definition of set A , redundancies in the equivalencing of degrees-of-freedom are omitted. For example, a degree-of-freedom shared by three substructures need only be equivalenced between two pairs of substructures, not all three pair-wise combinations. The compatibility constraint equations (i.e., the equations that enforce compatibility between the substructures) are then defined in the following manner. The r^{th} compatibility constraint equation will depend on a degree-of-freedom in substructure j if either the first or second element of the r^{th} 4-tuple in A is j . These compatibility constraint equations are expressed explicitly by defining the signed Boolean matrices \mathbf{B}_j for

$j = 1, \dots, M$ which have dimensions $n^I \times n_j$. Matrix \mathbf{B}_j has a 1 at location (r, p) if the r^{th} 4-tuple in A has the components (j, \cdot, p, \cdot) , and it has a -1 at location (r, q) if the r^{th} 4-tuple in A has the components (\cdot, j, \cdot, q) . Otherwise, the entries in matrix \mathbf{B}_j are zero. Thus, the compatibility constraint equations for all the substructures are symbolically represented by the system of n^I equations:

$$\sum_{i=1}^M \mathbf{B}_i \mathbf{u}_i = \mathbf{0} \quad (3.31)$$

The definition of the Boolean matrices \mathbf{B}_j in the present section reduces to the previous definition of the Boolean matrices for the case of only two substructures.

Utilizing these definitions, the minimization problem for the structural response given by statement (3.21) generalizes to:

$$\begin{aligned} & \min_{\substack{(\mathbf{u}_1, \dots, \mathbf{u}_{M-m}, \\ \mathbf{U}_{M-m+1}, \dots, \mathbf{U}_M, \\ \boldsymbol{\alpha}_{M-m+1}, \dots, \boldsymbol{\alpha}_M)}} \sum_{i=1}^{M-m} \left(\frac{1}{2} \mathbf{u}_i^T \mathbf{K}_i \mathbf{u}_i - \mathbf{F}_i^T \mathbf{u}_i \right) \\ & + \sum_{i=M-m+1}^M \left(\frac{1}{2} \mathbf{U}_i^T \mathbf{K}_i \mathbf{U}_i - \mathbf{F}_i^T (\mathbf{U}_i + \mathbf{R}_i \boldsymbol{\alpha}_i) \right) \\ \text{subject to: } & \sum_{i=1}^{M-m} \mathbf{B}_i \mathbf{u}_i + \sum_{j=M-m+1}^M \mathbf{B}_j (\mathbf{U}_j + \mathbf{R}_j \boldsymbol{\alpha}_j) = \mathbf{0} \\ & \mathbf{R}_i \mathbf{U}_i = \mathbf{0} \quad \text{for } i = M-m+1, \dots, M \end{aligned} \quad (3.32)$$

The necessary conditions for the minimization problem given by statement (3.32) are

$$\begin{aligned} & \mathbf{K}_i \mathbf{u}_i - \mathbf{F}_i + \mathbf{B}_i^T \boldsymbol{\lambda} = \mathbf{0} \quad \text{for } i = 1, \dots, M-m \\ & \tilde{\mathbf{K}}_i \left[\frac{\mathbf{U}_i}{\boldsymbol{\mu}_i} \right] - \left[\frac{\mathbf{F}_i - \mathbf{B}_i^T \boldsymbol{\lambda}}{\mathbf{0}} \right] = \mathbf{0} \quad \text{for } i = M-m+1, \dots, M \\ & \mathbf{R}_i^T (\mathbf{F}_i - \mathbf{B}_i^T \boldsymbol{\lambda}) = \mathbf{0} \quad \text{for } i = M-m+1, \dots, M \end{aligned} \quad (3.33)$$

along with the first constraint equation given in statement (3.32). Here λ are the Lagrange multipliers corresponding to the first constraint equation in statement (3.32), μ_i are the Lagrange multipliers for the last constraint equations in statement (3.32), and the matrices $\tilde{\mathbf{K}}_i$ for $i = M - m + 1, \dots, M$ are the augmented stiffness matrices given by

$$\tilde{\mathbf{K}}_i = \left[\begin{array}{c|c} \mathbf{K}_i & \mathbf{R}_i \\ \hline \mathbf{R}_i^T & [0] \end{array} \right] \quad (3.34)$$

The second equation in necessary conditions (3.33) combined with the conditions $\mathbf{R}_i^T \mathbf{K}_i = [0]$ for $i = M - m + 1, \dots, M$, and the fact that the columns of \mathbf{R}_i are linearly independent implies that the last equation in necessary conditions (3.33) can be replaced with the condition that the Lagrange multipliers μ_i are equal to zero. The structural responses \mathbf{u}_i for $i = 1, \dots, M - m$, and (\mathbf{U}_i, μ_i) for $i = M - m + 1, \dots, M$ are found by solving the first and second equations of necessary conditions (3.33), respectively, after the vector λ is determined.

The vector λ may be determined simultaneously with the rigid-body displacements α_i for $i = M - m + 1, \dots, M$ by a coordination subproblem. A coordination equation may be found by substituting the symbolic solutions for \mathbf{u}_i and \mathbf{U}_i into the compatibility constraint equation in statement (3.32), and solving this equation simultaneously with the last equation in necessary conditions (3.33). The following matrix equation gives the coordination subproblem:

$$\mathbf{M} \begin{bmatrix} \lambda \\ \alpha_{M-m+1} \\ \alpha_{M-m+2} \\ \vdots \\ \alpha_M \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{M-m} \mathbf{B}_i \mathbf{K}_i^{-1} \mathbf{F}_i + \sum_{i=M-m+1}^M [\mathbf{B}_i \mid [0]] \tilde{\mathbf{K}}_i^{-1} [\mathbf{F}_i^T \mid \mathbf{0}^T]^T \\ -\mathbf{R}_{M-m+1}^T \mathbf{F}_{M-m+1} \\ -\mathbf{R}_{M-m+2}^T \mathbf{F}_{M-m+2} \\ \vdots \\ -\mathbf{R}_M^T \mathbf{F}_M \end{bmatrix} \quad (3.35)$$

where the matrix \mathbf{M} is given by

$$\begin{bmatrix} \sum_{i=1}^{M-m} \mathbf{B}_i \mathbf{K}_i^{-1} \mathbf{B}_i^T + \sum_{i=M-m+1}^M [\mathbf{B}_i \mid [0]] \tilde{\mathbf{K}}_i^{-1} [\mathbf{B}_i \mid [0]]^T & -\mathbf{B}_{M-m+1} \mathbf{R}_{M-m+1} & \cdots & -\mathbf{B}_M \mathbf{R}_M \\ -(\mathbf{B}_{M-m+1} \mathbf{R}_{M-m+1})^T & [0] & \cdots & [0] \\ -(\mathbf{B}_{M-m+2} \mathbf{R}_{M-m+2})^T & [0] & \cdots & [0] \\ \vdots & \vdots & \ddots & \vdots \\ -(\mathbf{B}_M \mathbf{R}_M)^T & [0] & \cdots & [0] \end{bmatrix} \quad (3.36)$$

Statements (3.35) and (3.36) are the generalizations of statements (3.29) and (3.30) for multiple substructures. These statements are given for completeness, but are not utilized in the following design formulation.

Decomposition of the Structural-Design Problem

The starting point for the design decomposition is a simultaneous analysis and design formulation. The constraints are assumed to have the form $\mathbf{g}_i(\mathbf{v}_i, \mathbf{u}_i)$ for $i = 1, \dots, M$ in this derivation. The simultaneous analysis and design formulation is given by

$$\begin{aligned} & \min_{\substack{(\mathbf{v}_1, \dots, \mathbf{v}_M, \\ \mathbf{u}_1, \dots, \mathbf{u}_{M-m}, \\ \mathbf{U}_{M-m+1}, \dots, \mathbf{U}_M, \\ \alpha_{M-m+1}, \dots, \alpha_M, \\ \lambda, \mu_{M-m+1}, \dots, \mu_M)}} \sum_{i=1}^M W_i(\mathbf{v}_i) \\ & \text{subject to: } \mathbf{g}_i(\mathbf{v}_i, \mathbf{u}_i) \leq \mathbf{0} \quad \text{for } i = 1, \dots, M-m \end{aligned}$$

$$\begin{aligned}
\mathbf{g}_i(\mathbf{v}_i, \mathbf{U}_i + \mathbf{R}_i \boldsymbol{\alpha}_i) &\leq 0 & \text{for } i = M - m + 1, \dots, M \\
\mathbf{K}_i(\mathbf{v}_i) \mathbf{u}_i - \mathbf{F}_i + \mathbf{B}_i^T \boldsymbol{\lambda} &= 0 & \text{for } i = 1, \dots, M - m \\
\tilde{\mathbf{K}}_i(\mathbf{v}_i) \left[\frac{\mathbf{U}_i}{\boldsymbol{\mu}_i} \right] - \left[\frac{\mathbf{F}_i - \mathbf{B}_i^T \boldsymbol{\lambda}}{0} \right] &= 0 & \text{for } i = M - m + 1, \dots, M \\
\sum_{i=1}^{M-m} \mathbf{B}_i \mathbf{u}_i + \sum_{j=M-m+1}^M \mathbf{B}_j (\mathbf{U}_j + \mathbf{R}_j \boldsymbol{\alpha}_j) &= 0 & (3.37) \\
\mathbf{R}_i^T (\mathbf{F}_i - \mathbf{B}_i^T \boldsymbol{\lambda}) &= 0 & \text{for } i = M - m + 1, \dots, M
\end{aligned}$$

This expression is simplified by the following steps: 1) specify that \mathbf{u}_i for $i = 1, \dots, M - m$ and \mathbf{U}_i for $i = M - m + 1, \dots, M$ (which are implicit functions of \mathbf{v}_i and $\boldsymbol{\lambda}$) are known from solutions of structural-response subproblems (such as problems (3.19) and (3.22)), and thus identically satisfy the first two equality constraints in statement (3.37); and 2) for the remaining functions in (3.37), replace \mathbf{g}_i , \mathbf{u}_i , and \mathbf{U}_i with their first-order Taylor series approximations about the point $(\mathbf{v}_i, \boldsymbol{\alpha}_i, \boldsymbol{\lambda})$. The design variables for the structural-sizing subproblem are given by $\mathbf{x}_0 \equiv (\mathbf{v}_1, \dots, \mathbf{v}_M, \boldsymbol{\alpha}_{M-m+1}, \dots, \boldsymbol{\alpha}_M, \boldsymbol{\lambda})$. These simplifications reduce statement (3.37) to the following structural-sizing subproblem

$$\begin{aligned}
&\min_{\mathbf{x}_0 \equiv (\mathbf{v}_1, \dots, \mathbf{v}_M, \boldsymbol{\alpha}_{M-m+1}, \dots, \boldsymbol{\alpha}_M, \boldsymbol{\lambda})} \sum_{i=1}^M W_i(\mathbf{v}_i) \\
&\text{subject to: } \mathbf{v}_i^l \leq \mathbf{v}_i \leq \mathbf{v}_i^u & \text{for } i = 1, \dots, M \\
&\mathbf{g}_i + \left. \frac{\partial \mathbf{g}_i}{\partial \mathbf{v}_i} \right|_{\boldsymbol{\lambda}} (\mathbf{v}_i - \mathbf{v}_i^0) + \left. \frac{\partial \mathbf{g}_i}{\partial \boldsymbol{\lambda}} \right|_{\mathbf{v}_i} (\boldsymbol{\lambda} - \boldsymbol{\lambda}_i) \leq 0 & \text{for } i = 1, \dots, M - m \\
&\mathbf{g}_i + \left. \frac{\partial \mathbf{g}_i}{\partial \mathbf{v}_i} \right|_{\boldsymbol{\lambda}, \boldsymbol{\alpha}_i} (\mathbf{v}_i - \mathbf{v}_i^0) + \left. \frac{\partial \mathbf{g}_i}{\partial \boldsymbol{\lambda}} \right|_{\mathbf{v}_i, \boldsymbol{\alpha}_i} (\boldsymbol{\lambda} - \boldsymbol{\lambda}_i) \\
&\quad + \left. \frac{\partial \mathbf{g}_i}{\partial \boldsymbol{\alpha}_i} \right|_{\mathbf{v}_i, \boldsymbol{\lambda}} (\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_i^0) \leq 0 & \text{for } i = M - m + 1, \dots, M \\
&\sum_{i=1}^{M-m} \mathbf{B}_i \left(\mathbf{u}_i + \left. \frac{\partial \mathbf{u}_i}{\partial \mathbf{v}_i} \right|_{\boldsymbol{\lambda}} (\mathbf{v}_i - \mathbf{v}_i^0) + \left. \frac{\partial \mathbf{u}_i}{\partial \boldsymbol{\lambda}} \right|_{\mathbf{v}_i} (\boldsymbol{\lambda} - \boldsymbol{\lambda}_i) \right) & (3.38)
\end{aligned}$$

$$\begin{aligned}
& + \sum_{j=M-m+1}^M \mathbf{B}_j \left(\mathbf{U}_j + \frac{\partial \mathbf{U}_j}{\partial \mathbf{v}_j} \Big|_{\boldsymbol{\lambda}} (\mathbf{v}_j - \mathbf{v}_j^0) \right. \\
& \quad \left. + \frac{\partial \mathbf{U}_j}{\partial \boldsymbol{\lambda}} \Big|_{\mathbf{v}_i} (\boldsymbol{\lambda} - \boldsymbol{\lambda}_j) + \mathbf{R}_j \boldsymbol{\alpha}_j \right) = 0 \\
& \quad \mathbf{R}_i^T (\mathbf{F}_i - \mathbf{B}_i^T \boldsymbol{\lambda}) = 0 \quad \text{for } i = M - m + 1, \dots, M
\end{aligned}$$

where move limits \mathbf{v}_i^l and \mathbf{v}_i^u that are adjusted during the solution process have been introduced to ensure convergence. In this subproblem, the design variables $\boldsymbol{\lambda}$ are coupling variables, and the compatibility constraints (the next to last constraint shown in statement (3.38)) are coupling constraints. Because of the relatively small number of these coupling variables and coupling constraints, a further decomposition of this structural-sizing subproblem into smaller subproblems could be attempted using the method of Ritter ([29], pp. 276–283), or the method of Ha [18]. However, no further decomposition of the structural-sizing subproblem is undertaken in the present dissertation.

The structural-response subproblems, one for each substructure, determine the structural displacements, the design constraints, and their derivatives with respect to the elements of \mathbf{x}_0 . The design variables of the structural-response subproblems for substructures $i = 1, \dots, M - m$ are $\mathbf{x}_i \equiv (\mathbf{u}_i, \mathbf{v}_i^0, \boldsymbol{\lambda}_i, \partial \mathbf{u}_i / \partial \mathbf{v}_i |_{\boldsymbol{\lambda}}, \partial \mathbf{u}_i / \partial \boldsymbol{\lambda} |_{\mathbf{v}_i}, \mathbf{g}_i, \partial \mathbf{g}_i / \partial \mathbf{v}_i |_{\boldsymbol{\lambda}}, \partial \mathbf{g}_i / \partial \boldsymbol{\lambda} |_{\mathbf{v}_i})$. The necessary condition form of these subproblems is

$$\begin{aligned}
\mathbf{K}_i \mathbf{u}_i &= \mathbf{F}_i - \mathbf{B}_i^T \boldsymbol{\lambda} \\
\mathbf{v}_i^0 &= \mathbf{v}_i \\
\boldsymbol{\lambda}_i &= \boldsymbol{\lambda} \\
\mathbf{g}_i &= \mathbf{g}_i(\mathbf{v}_i, \mathbf{u}_i)
\end{aligned}$$

$$\begin{aligned}
\mathbf{K}_i \frac{\partial \mathbf{u}_i}{\partial \mathbf{v}_i} \Big|_{\boldsymbol{\lambda}} &= -\frac{\partial \mathbf{K}_i \mathbf{u}_i}{\partial \mathbf{v}_i} \\
\mathbf{K}_i \frac{\partial \mathbf{u}_i}{\partial \boldsymbol{\lambda}} \Big|_{\mathbf{v}_i} &= -\mathbf{B}_i^T \\
\frac{\partial g_{i,j}}{\partial v_{i,k}} \Big|_{\boldsymbol{\lambda}} &= \frac{g_{i,j} \left((\mathbf{v}_i + \mathbf{e}_k^{\mathbf{v}_i} \Delta v_{i,k}), (\mathbf{u}_i + \partial \mathbf{u}_i / \partial \mathbf{v}_i \Big|_{\boldsymbol{\lambda}} \mathbf{e}_k^{\mathbf{v}_i} \Delta v_{i,k}) \right) - g_{i,j}}{\Delta v_{i,k}} \\
\frac{\partial g_{i,j}}{\partial \lambda_k} \Big|_{\mathbf{v}_i} &= \frac{g_{i,j} \left(\mathbf{v}_i, (\mathbf{u}_i + \partial \mathbf{u}_i / \partial \boldsymbol{\lambda} \Big|_{\mathbf{v}_i} \mathbf{e}_k^{\boldsymbol{\lambda}} \Delta \lambda_k) \right) - g_{i,j}}{\Delta \lambda_k}
\end{aligned} \tag{3.39}$$

for $i = 1, \dots, M - m$. The design variables of the structural-response subproblems for substructures $i = M - m + 1, \dots, M$ are $\mathbf{x}_i \equiv (\mathbf{U}_i, \mathbf{v}_i^0, \boldsymbol{\lambda}_i, \boldsymbol{\alpha}_i^0, \partial \mathbf{U}_i / \partial \mathbf{v}_i \Big|_{\boldsymbol{\lambda}}, \partial \mathbf{U}_i / \partial \boldsymbol{\lambda} \Big|_{\mathbf{v}_i}, \mathbf{g}_i, \partial \mathbf{g}_i / \partial \mathbf{v}_i \Big|_{\boldsymbol{\lambda}, \boldsymbol{\alpha}_i}, \partial \mathbf{g}_i / \partial \boldsymbol{\lambda} \Big|_{\mathbf{v}_i, \boldsymbol{\alpha}_i}, \partial \mathbf{g}_i / \partial \boldsymbol{\alpha} \Big|_{\mathbf{v}_i, \boldsymbol{\lambda}})$. The necessary condition form of these subproblems is

$$\begin{aligned}
\tilde{\mathbf{K}}_i \begin{bmatrix} \mathbf{U}_i \\ \boldsymbol{\mu}_i \end{bmatrix} &= \begin{bmatrix} \mathbf{F}_i - \mathbf{B}_i^T \boldsymbol{\lambda} \\ \mathbf{0} \end{bmatrix} \\
\mathbf{v}_i^0 &= \mathbf{v}_i \\
\boldsymbol{\lambda}_i &= \boldsymbol{\lambda} \\
\boldsymbol{\alpha}_i^0 &= \boldsymbol{\alpha}_i \\
\mathbf{g}_i &= \mathbf{g}_i(\mathbf{v}_i, \mathbf{U}_i + \mathbf{R}_i \boldsymbol{\alpha}_i) \\
\tilde{\mathbf{K}}_i \begin{bmatrix} \frac{\partial \mathbf{U}_i}{\partial \mathbf{v}_i} \Big|_{\boldsymbol{\lambda}} \\ \frac{\partial \boldsymbol{\mu}_i}{\partial \mathbf{v}_i} \Big|_{\boldsymbol{\lambda}} \end{bmatrix} &= -\frac{\partial \tilde{\mathbf{K}}_i [\mathbf{U}_i^T \mid \mathbf{0}^T]^T}{\partial \mathbf{v}_i} \\
\tilde{\mathbf{K}}_i \begin{bmatrix} \frac{\partial \mathbf{U}_i}{\partial \boldsymbol{\lambda}} \Big|_{\mathbf{v}_i} \\ \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\lambda}} \Big|_{\mathbf{v}_i} \end{bmatrix} &= -\begin{bmatrix} \mathbf{B}_i^T \\ \mathbf{0} \end{bmatrix} \\
\frac{\partial g_{i,j}}{\partial v_{i,k}} \Big|_{\boldsymbol{\lambda}, \boldsymbol{\alpha}_i} &= \frac{g_{i,j} \left((\mathbf{v}_i + \mathbf{e}_k^{\mathbf{v}_i} \Delta v_{i,k}), (\mathbf{U}_i + \partial \mathbf{U}_i / \partial \mathbf{v}_i \Big|_{\boldsymbol{\lambda}} \mathbf{e}_k^{\mathbf{v}_i} \Delta v_{i,k} + \mathbf{R}_i \boldsymbol{\alpha}_i) \right) - g_{i,j}}{\Delta v_{i,k}} \\
\frac{\partial g_{i,j}}{\partial \lambda_k} \Big|_{\mathbf{v}_i, \boldsymbol{\alpha}_i} &= \frac{g_{i,j} \left(\mathbf{v}_i, (\mathbf{U}_i + \partial \mathbf{U}_i / \partial \boldsymbol{\lambda} \Big|_{\mathbf{v}_i} \mathbf{e}_k^{\boldsymbol{\lambda}} \Delta \lambda_k + \mathbf{R}_i \boldsymbol{\alpha}_i) \right) - g_{i,j}}{\Delta \lambda_k}
\end{aligned} \tag{3.40}$$

$$\left. \frac{\partial g_{i,j}}{\partial \alpha_{i,k}} \right|_{\mathbf{v}_i, \boldsymbol{\lambda}} = \frac{g_{i,j}(\mathbf{v}_i, (\mathbf{U}_i + \mathbf{R}_i(\boldsymbol{\alpha}_i + \mathbf{e}_k^\alpha \Delta \alpha_{i,k}))) - g_{i,j}}{\Delta \alpha_{i,k}}$$

for $i = M - m + 1, \dots, M$. Several features of the subproblems defined by statements (3.39) and (3.40) should be noted. The design variables \mathbf{v}_i^0 , $\boldsymbol{\alpha}_i^0$, and $\boldsymbol{\lambda}_i$ preserve the values of the structural-sizing variables at which the sensitivity analyses are performed for use in the structural-sizing subproblem. The sensitivities of the substructure displacements with respect to the sizing variables \mathbf{v}_i are obtained by differentiating the first two equations in statement (3.33) with respect to the sizing variables in the same manner that the sensitivities in statement (3.10) are obtained. However, the computational effort is greatly decreased when using substructures since both the order of the systems to be solved and the number of right hand sides (equal to the number of substructure sizing variables) to be formed and evaluated are reduced. The derivatives with respect to the Lagrange multipliers $\boldsymbol{\lambda}$ are obtained by differentiating the first two equations in statement (3.33). The cost for performing these sensitivity analyses is also nominal because the order of the equations to be solved is small, the right hand side contains many zero columns (the nonzero columns are equal in number to the number of degrees of freedom that are equivalenced in the substructure), and this right hand side needs to be formed only once. Finally, the constraint derivatives are found using finite differences that incorporate the displacement sensitivity derivatives, but if $g_{i,j}$ is a stress or buckling constraint then $\partial g_{i,j} / \partial \alpha_{i,k} |_{\mathbf{v}_i, \boldsymbol{\lambda}}$ does not need to be computed since it is identically zero. As a point of interest, the approximate sensitivity derivative formulation of the previous section (i.e., design formulation 3) could be utilized to replace the sensitivity derivatives in statements (3.39) and (3.40) during the solution

process to provide further computational gains. However, this approach is not pursued in the present dissertation.

Chapter 4

Numerical Results for Equilibrium Programming Structural-Design Formulations

The equilibrium programming design formulations derived in the previous chapter appear to offer significant computational benefits, and these benefits are verified by example problems in the present chapter. Since design formulations 1 and 2 are essentially methods currently utilized in structural design, only formulations 3 and 4 are investigated in the example problems. The results using these formulations are compared with results using a conventional optimization approach which is actually an implementation of formulation 2. In subsequent sections, the implementation of the analysis and optimization algorithms is outlined, the example problems are described, and the results of using formulations 3 and 4 on these example problems are compared with results using the conventional approach.

4.1 Implementation of Numerical Algorithms

In this section, general implementation issues common to all the formulations are discussed first. Then implementation details specific to formulations 3 and 4 are described.

4.1.1 General Implementation Issues

For practical structural-design problems, an equilibrium-programming-based solution algorithm should utilize existing finite element structural-analysis software. The sensitivity derivative runstreams of [9] were modified, and several FORTRAN routines were incorporated into the Engineering Analysis Language (EAL) structural-analysis code [66] to solve the example design problems described subsequently. Both a conventional solution approach and the equilibrium programming design formulations are developed using this software infrastructure. The sensitivity derivative runstreams utilize the semi-analytic method, originally developed in [14], to compute the displacement sensitivity derivatives. Derivatives of the stress and local buckling constraints are calculated using these displacement derivatives by a finite-difference approach. An example of the finite-difference derivatives for general constraints is given by the following expression:

$$\frac{\partial \mathbf{g}}{\partial v_j} = \frac{\mathbf{g}(\mathbf{v} + \Delta v_j \mathbf{e}_j^v, \mathbf{u} + \partial \mathbf{u} / \partial v_j \Delta v_j \mathbf{e}_j^v) - \mathbf{g}(\mathbf{v}, \mathbf{u})}{\Delta v_j} \quad (4.1)$$

The sequential solution approach described in a previous chapter is the solution approach chosen for all the examples in the present dissertation, but estimates of the speedup due to parallelization of subproblems are given for EP structural design formulation 4. In the sequential solution approach, the structural-sizing subproblem and the structural-response subproblems are solved alternately in a cyclical manner until the solutions from consecutive cycles converge. In the present dissertation, the nonlinear, structural-sizing subproblem is replaced with a linear programming approximation, which is an approach that has proven to be very robust. Thus, when results using a con-

ventional solution approach are discussed, this approach is sequential linear programming (SLP).

The linear programming method utilized is based on the L_1 penalized objective function method of [70]. In this reference, the objective function W is replaced by a penalized objective function P given by

$$P = W + K \sum_i \max(g_i, 0) \quad (4.2)$$

where K is a constant that must be larger than the largest of all the Lagrange multipliers of the constraint function vector \mathbf{g} . In the penalized objective function method, the maximum function in (4.2) is implemented by using an additional positive design variable as a slack variable for each constraint. Each slack variable is subtracted from its constraint, and also used in place of the maximum function in equation (4.2). This penalty method ensures that if no feasible solution is possible, a solution is found that will minimize the magnitude of the vector of violated constraints using the L_1 (i.e., the sum of the absolute values) norm. To ensure that the linear programming approximation of a subproblem has sufficient accuracy, move limits are used as additional side constraints on the sizing variables. Move limits are easily incorporated within the EP theory given in [68]. In the present dissertation, the move limits on the design variables are computed by limiting the magnitude of the allowable change to a factor times the design variable values. A move-limit-factor control strategy is utilized in the example problems described subsequently. The move-limit factor is typically initialized with a value of 10%. This factor is reduced by half whenever a criterion indicates a reduction is necessary. The criterion used in the present study stipulates that the move-limit factor is reduced when

the penalized objective function (4.2), calculated exactly using the latest design information, increases from the previously calculated value. No method to increase the move-limit factor, such as given in [70], is used. Thus, the resulting design history is somewhat sensitive to the value assumed for K . For very large values of K , the optimization procedure favors remaining in the feasible region over minimizing the objective function. In practice, using a very large value of K leads to a premature reduction in the move limits during the design process, and slows the convergence to the final design. In the reported results, the value assumed for K is approximately twice the value of the largest Lagrange multiplier of the design constraints obtained during the design history. The specific routine incorporated within EAL to solve the linear programming problems is the sparse MINOS routine [33].

For some of the larger structural models studied, there can be thousands of elements that must be examined for local stress and buckling constraints, and these elements must be optimally sized. To reduce the number of design variables, the design variable linking method of [39] is utilized. Specifically, the finite element model is partitioned into regions, and within each region, all the elements are sized by the same design variables. In addition, to reduce the number of constraints, the local constraints within each region are “lumped” using the Kresselmeier-Steinhauser cumulative constraint [27]. These reduction methods are utilized only for the civil transport problems described subsequently.

4.1.2 Implementation Issues for Design Formulation 3

The flowchart in figure 4.1 illustrates a computer implementation of formulation 3 for solving a structural-design problem. An iteration is represented by a circuit through outermost loop in the figure. In the results to be described subsequently, the iterations shown are described as either approximate or exact. An approximate iteration is defined as a pass through the outer loop of figure 4.1 utilizing the approximate sensitivity derivatives; an exact iteration is defined as a pass through this outer loop utilizing exact sensitivity derivatives. The solution method for formulation 3 is similar to that of conventional approximation-based design methods, such as the SLP method that is used for comparisons in the results section, except for the logic shown between the boxes labeled “Exact Structural Response and Constraints” and “Structural-Sizing Subproblem.” In a flowchart for a conventional approximation-based method, the “No” branch of the “ P Increased?” decision box would simply go to “Exact Sensitivity Derivatives,” and the “Yes” branch would go to “Reset Design” and then to “Reduce Move Limits.” Because the iterations utilizing approximate sensitivity derivatives are so much less expensive than those utilizing exact sensitivity derivatives, the most computationally efficient procedure would utilize approximate constraint sensitivity derivatives as often as possible, and would calculate exact constraint sensitivity derivatives only when the inaccuracy of the approximate sensitivity derivatives inhibits convergence. This approach is utilized in figure 4.1 where the criterion for choosing to update the sensitivity derivatives instead of calculating exact sensitivity derivatives is based on the behavior of the L_1 penalized objective function of the structural-sizing subproblem.

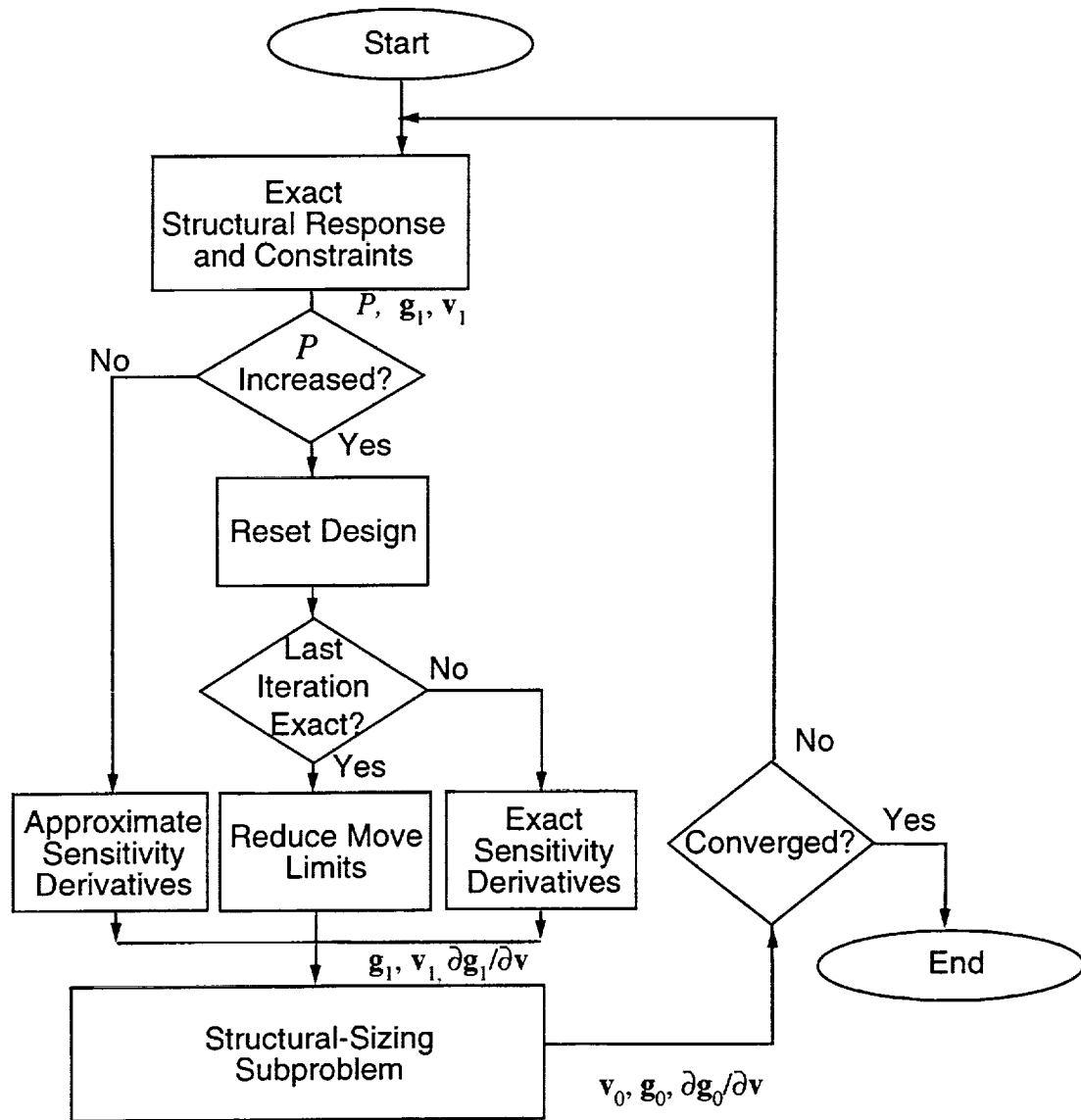


Figure 4.1: Flowchart of the solution procedure for structural design formulation 3.

The penalized objective function is also utilized in the strategy for adjusting the move-limit-control factor in a variation of the approach described in the previous subsection. If the value of P given by equation (4.2) (and computed within the box labeled “Exact Structural Response and Constraints” in figure 4.1) decreases after an approximate iteration, that iteration is accepted. If the value of P increases after an approximate iteration, the iteration is rejected and exact sensitivity derivatives are calculated at the design conditions existing at the beginning of the rejected approximate iteration. Thus, an exact iteration begins. If the value of P increases after an exact iteration, the results of the exact iteration are rejected, the move-limit factor is decreased by half, and the exact iteration is restarted using the design conditions existing at the beginning of the rejected exact iteration. The costs of a rejected iteration are the expenses of calculating solutions to subproblems (3.14) and (3.15) where no sensitivity calculations are performed for the latter subproblem. Both these expenses are minor compared to the cost of an exact sensitivity analysis when the structural-design problem is large.

4.1.3 Implementation Issues for Design Formulation 4

A flowchart for the implementation of design formulation 4 would be very similar to that for a conventional approximation-based method as described in the previous subsection. However, instead of performing analysis and sensitivity calculations for a single structure, the analyses and sensitivity calculations for n substructures are performed. The penalized objective function incorporates only the inequality constraints, and no slack variables are used with the equality constraints in the structural-sizing subproblem. In the example problem

for design formulation 4, the structural-response subproblems for the substructures are solved in sequence, although they could be solved in parallel. The sublibrary capability of EAL is used extensively in solving the subproblems so that the data associated with each subproblem are stored in separate libraries. In addition, the ability to manipulate matrix data blocks and to add extra terms to matrices is utilized in forming the augmented stiffness matrices. However, because the augmented stiffness matrices are nonpositive definite, some experimentation in determining the order in which the degrees of freedom are eliminated in the matrix factorization is necessary to avoid obtaining erroneous singular matrix messages. Typically, the degrees of freedom for the vectors μ_i would have to be eliminated before completing the elimination of the degrees of freedom for the vectors U_i .

4.2 Description of the Example Problems

The four example problems used to evaluate the equilibrium programming design formulations are described. The first three example problems are utilized only with design formulation 3, and the final example problem is used only with design formulation 4. The first example problem is the common ten-bar-truss weight optimization problem that is described in several references. The second example problem is a more complex high-speed transport wing weight optimization problem, and the third example problem is the weight optimization of a half-symmetric model for an entire high-speed transport vehicle. The use of updated sensitivity derivatives on the first two example problems has been investigated by the author previously [52], but the solution method used for the present results is an improvement over that of reference [52]. The constraints

for the first structural optimization example include stress, displacement and minimum gauge constraints. The second and third example problems include additional local buckling constraints, but the third example problem does not include displacement constraints. This third example is the largest problem investigated in the present dissertation both in terms of the number of design variables and the number of displacement degrees of freedom. The fourth example problem is the minimum weight design for a transmission tower that is decomposed into substructures in two ways. This example problem uses only stress and minimum gauge design constraints. These problems are described in more detail in subsequent subsections.

4.2.1 Ten-Bar-Truss Example Problem

The minimum weight ten-bar-truss example problem is illustrated in figure 4.2. This problem is described in [19] (p. 244), but a brief description is repeated here for completeness. The vertical and horizontal members are each 360 inches in length. The material properties assumed are those for aluminum with a failure stress of 37,500 psi, a Young's modulus of 10^7 psi, a Poisson's ratio of 0.3, and a density of 0.1 lb/in³. Two 100,000-lb loads are applied, as shown in figure 4.2, and the upper displacement limits $\delta_1 = \delta_2 = 2.0$ in. for the displacement constraints are shown in the figure. The design variables are the cross-sectional areas of the numbered bars in figure 4.2 which all have the initial value 10 in² that yields an infeasible initial design. The minimum gauge assumed is 0.1 in², and the penalty coefficient K utilized for this problem is 10,000 lb.

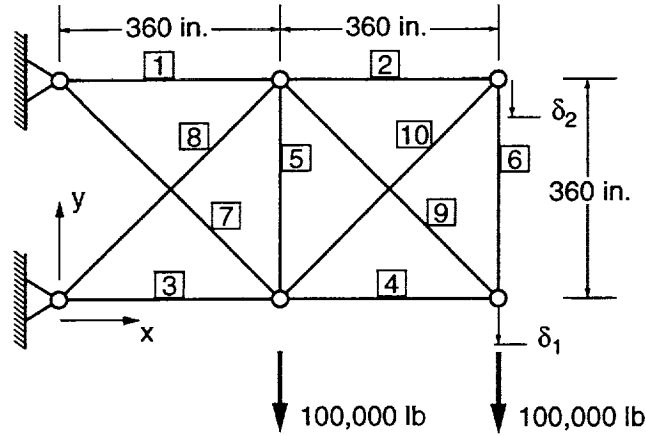


Figure 4.2: Schematic of geometry, loads, and displacement limits for the ten-bar-truss example problem.

4.2.2 High-Speed Civil Transport Wing Example Problem

The second structural optimization example problem considered for EP structural-design formulation 3 is the structural sizing of the wing for a proposed high-speed civil transport concept described in [43]. The details defining this structural-design problem are too numerous to list so only a summary of its features is presented. The finite element model of the wing is shown in figure 4.3. The upper wing cover panels are removed in this figure to illustrate the rib and spar web arrangement. The cover panels are titanium honeycomb-core sandwich panels, and the shear webs are titanium sine-wave webs. The model is relatively detailed with 1728 nodes (10,144 degrees of freedom) and 2447 elements. A single load condition is analyzed in the structural response which represents a 2.5g balanced, symmetric supersonic pull-up maneuver. There are 41 design variables considered in the structural optimization. These design variables include facesheet thicknesses for the sandwich panels, honeycomb-core heights, and sine-wave web gauges. The model uses a simple form of design variable

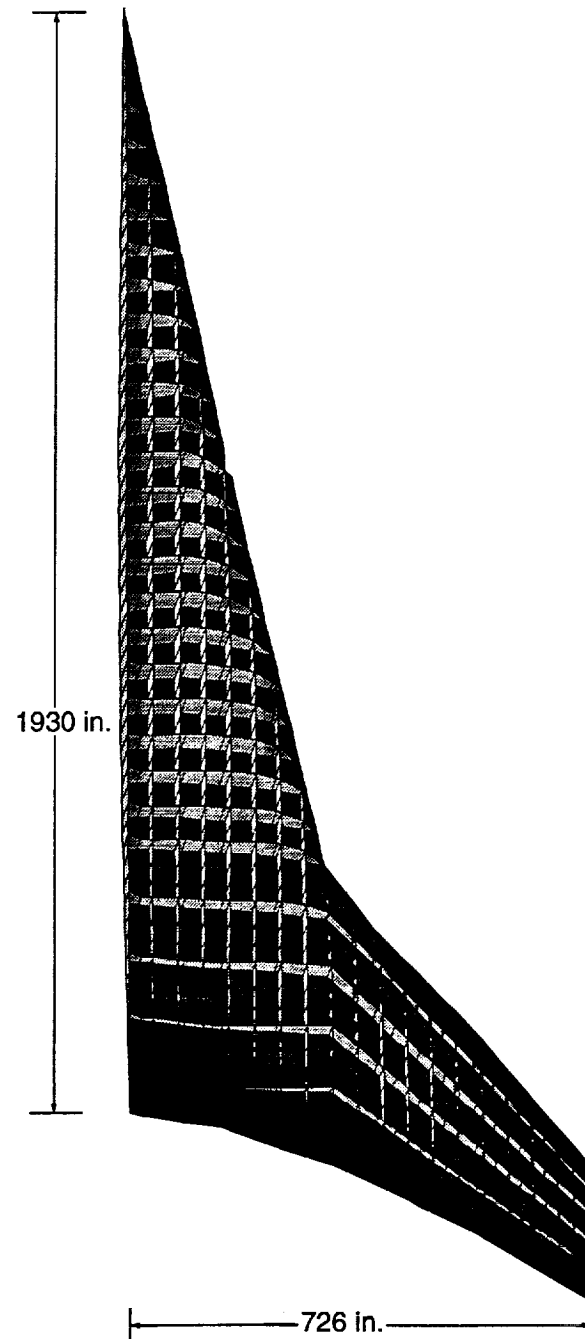


Figure 4.3: Finite element model of high-speed civil transport wing with upper cover panels removed.

linking such that each sizing variable controls multiple elements. There are 212 design constraints that are cumulative constraints over groups of elements, half of which are stress constraints and the remainder are local buckling constraints. The most critical constraint in this example is a 12-foot limit on maximum deflection of the wing tip. The initial design utilized in this problem is feasible. The penalty coefficient K utilized for this problem is 50,000 lb, and the value for the parameter in the Kreisselmeier-Steinhauser function [27] for the cumulative constraints is taken to be 50.

4.2.3 High-Speed Transport Vehicle Example Problem

A third example problem considered is the structural optimization of a half-symmetric model for an entire high-speed transport aircraft¹. The finite element model of the vehicle is shown in figure 4.4. The cover panels and webs are honeycomb-core sandwich panels having polymer-matrix composite facesheets. The model is very large with 7301 nodes (43,806 degrees of freedom) and 14,293 elements. Two load conditions are analyzed for the structural response: 2.5g and $-1.0g$ balanced, symmetric supersonic maneuvers. This model also uses design variable linking such that each sizing variable controls multiple elements in a design region. There are 348 design variables considered in the structural optimization. These design variables are the thickness of laminates in three major directions (i.e., in directions oriented 0° , 90° , and $\pm 45^\circ$ relative to the primary load paths), and the honeycomb-core heights. The constraints in this example are minimum gauge side constraints, and constraints

¹The structural model for this example has been supplied by the Boeing Company and the results are presented without absolute scales in this dissertation under the conditions of a NASA Langley Property Loan Agreement, Loan Control Number I922931.

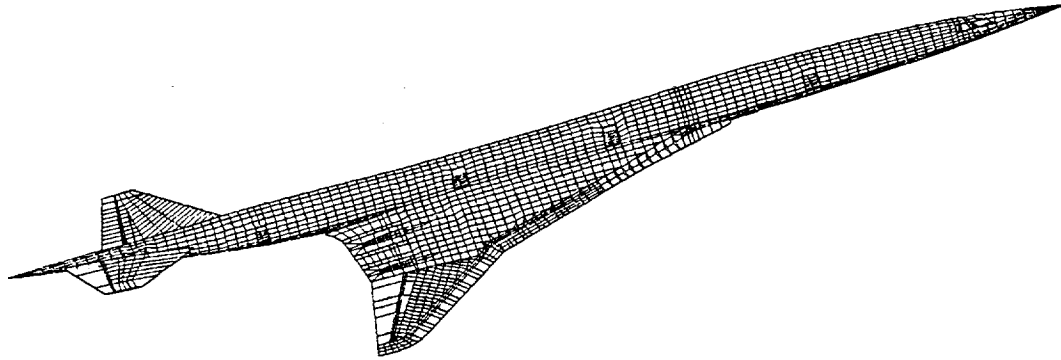


Figure 4.4: Half-symmetric, finite element model of the high-speed transport vehicle.

of strength and panel buckling for each element in the model. These strength and buckling constraints are incorporated into 260 cumulative constraints using the Kreisselmeier-Steinhauser function [27], one for each design region (the value of the parameter in the Kreisselmeier-Steinhauser function is 50). The initial design utilized in this problem is feasible, and the normalized penalty coefficient K utilized for this problem is 0.053.

4.2.4 Transmission Tower Example Problem

The fourth example problem is a transmission tower weight optimization. The geometry of the tower, the loading conditions, and the decomposition into substructures is shown in figure 4.5. Results are presented for a decomposition into two substructures (denoted as 1 and 2), and a decomposition into four substructures (denoted as I, II, III, and IV). This example is similar to the

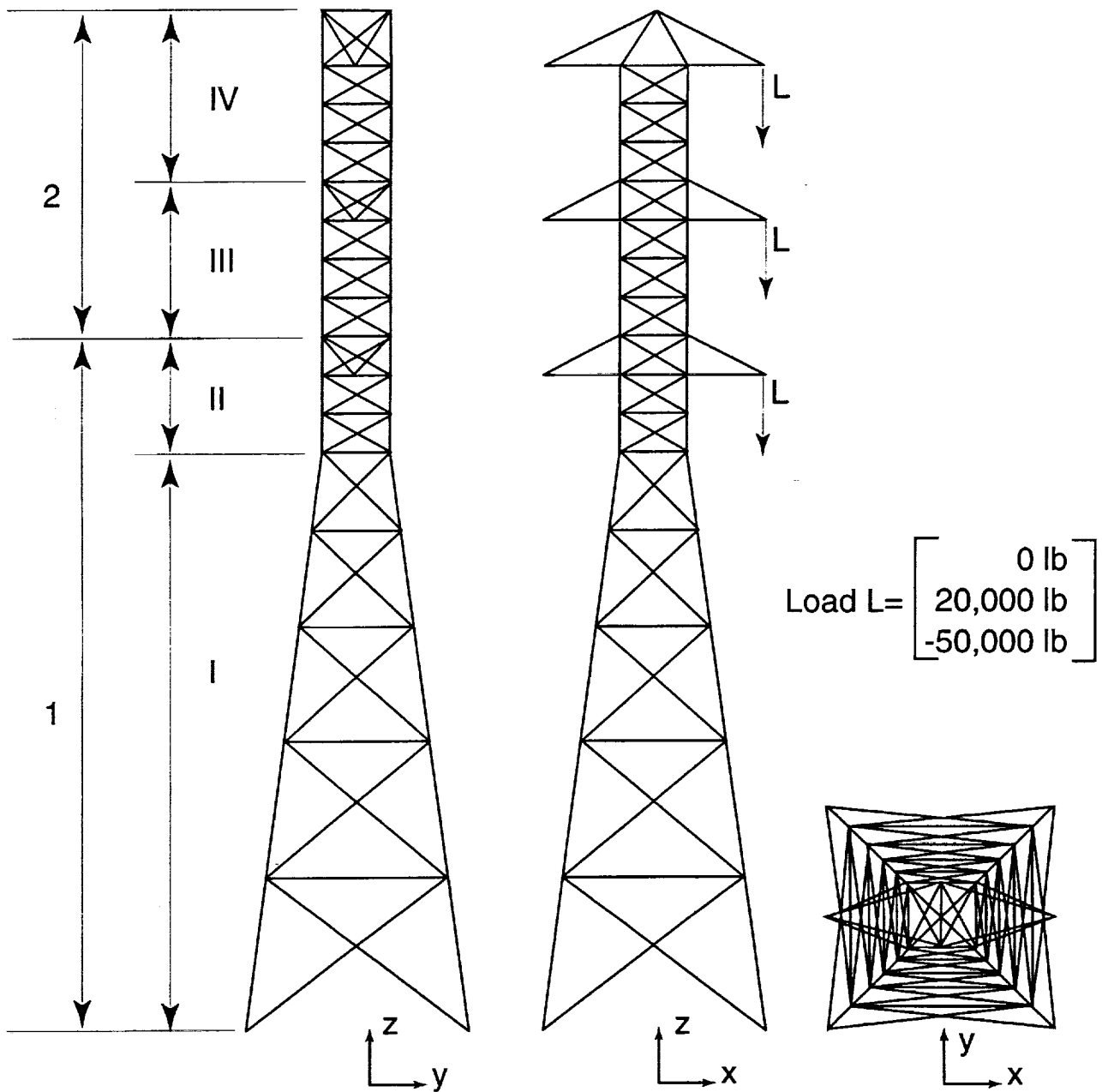


Figure 4.5: Schematic of the transmission tower example geometry, loading, and division into substructures.

tower example described in [36], and because of the small number of interface degrees of freedom, it is ideal for application of substructure techniques. The tower is 634.5 inches tall, and there are 72 nodes and 279 bars in the model. An independent design variable is used for each bar cross-sectional area. This approach creates a problem with 279 design variables by ignoring the symmetries in bar cross-sectional areas that would be present in a practical design (since the loads could be applied to either set of tower arms, and in either the positive or negative y direction). The material chosen for this example problem is aluminum with properties and minimum gauges as described previously. The cross-sectional areas of the bars all have the initial value 10 in^2 that yields a feasible initial design. Only stress inequality constraints are utilized for this example problem, and the value of the penalty coefficient K used is 160 lb.

4.3 Results for the Example Problems

The results for the four example problems follow. The design history of the penalized weight objective function, the weight objective function, and the most critical constraint are presented for each example. Also, normalized CPU time comparisons for each example problem are presented. The results for these problems have been computed using two types of RISC workstations.

4.3.1 Ten-Bar-Truss Example Problem Results

The CPU time required for one approximate iteration for the ten-bar-truss problem is approximately 16% of the time required for an exact iteration. This CPU time for an approximate iteration is only 21% larger than the cost of performing the structural response analysis and constraint evaluation alone.

Therefore, efficiency benefits are expected using the approximate sensitivity updates.

The convergence histories for the penalized objective function for 5 different cases are shown in figure 4.6. In this figure, results are shown using formulation 3 with initial move-limit factors of 20%, 10%, and 5%. The move-limit factors are denoted by ML_I in this and subsequent figures. In addition, results using a conventional SLP solution approach are shown for comparison with initial move-limit factors of 20% and 10%. The SLP results are identical to formulation 3 if all iterations are chosen to be exact. The solution procedure is halted when the move-limit factor is reduced below 1%. The final penalized objective functions and the final objective functions in all five design cases are within a pound of each other. The horizontal line shown in figure 4.6 shows a penalized objective function value 1% higher than the final value, and this penalized objective function value is used as the criterion for convergence. Because the initial design is infeasible, in all cases there is a rapid decrease in the penalized objective function as the violated constraints become satisfied. The 20%, 10%, and 5% cases using EP structural-design formulation 3 converge after 55, 92, and 101 iterations, respectively. This count of iterations is inflated slightly because it includes a rejected approximate iteration before every exact iteration after the first one. The 20% and 10% SLP cases converge at iterations 62 and 84, respectively. The large number of iterations required for all these cases results from the simple linearization of the design constraints utilized in the present approach. A more sophisticated constraint approximation scheme, such as the use of a linearization in reciprocal variables, would decrease the number of iterations to convergence, but at the cost of making the structural-

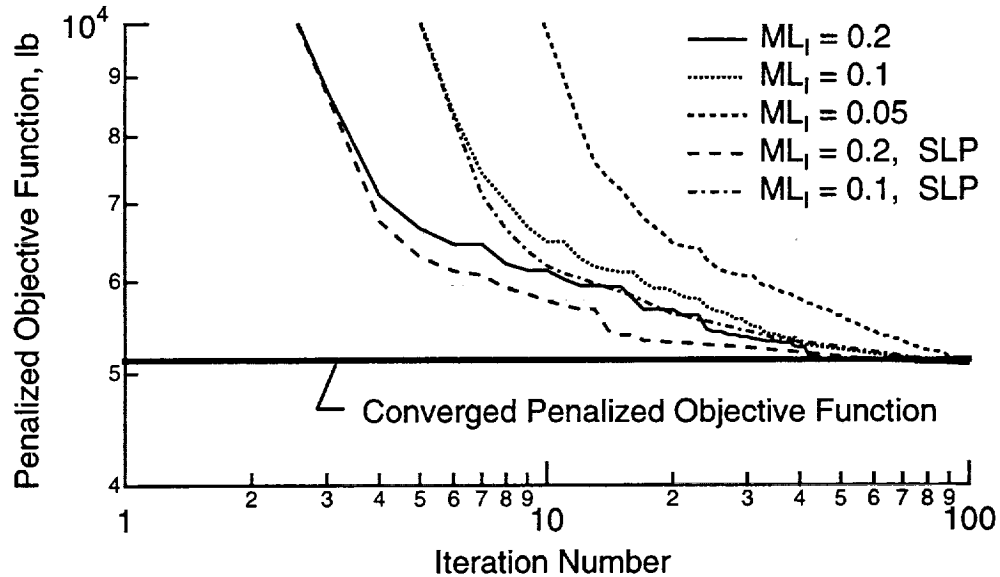


Figure 4.6: Comparison of penalized objective function iteration histories for the ten-bar truss with different initial move-limit-control factors using EP structural-design formulation 3, and using a conventional SLP approach.

sizing problem a NLP problem instead of a LP problem. A common trend for all these results is that there is an increase in the number of iterations required as the initial move-limit factor is decreased. It is somewhat surprising that the formulation 3 case with a 20% initial move-limit factor converged sooner than the comparable SLP case, because, for most of the design history, the penalized objective function is smaller for the latter case. This result occurs because the move-limit factor for the SLP case remains too large for most of the latter part of the iteration history, and it is expected that, in general, more iterations are required for formulation 3 to converge than for the comparable SLP method. The number of exact sensitivity calculations prior to convergence for the 20%, 10%, and 5% cases using formulation 3 are 17, 12, and 11, respectively. Thus, using the number of exact sensitivity analyses as an indicator of how much work is required to solve this problem, formulation 3 is a very efficient method.

The iteration histories of the objective function and the displacement constraint on δ_1 are shown in figures 4.7 and 4.8, respectively. Only the cases having initial move-limit factors of 20% and 10% are shown in these figures. In all the cases shown in figure 4.7, the objective function value rises initially because the optimizer is unable to satisfy the constraints in this initially infeasible design, as shown by figure 4.8, so it minimizes the penalized objective function described previously. The objective function results for formulation 3 are typically larger than the SLP results during the initial iterations when the move limits are large, but the results having the same initial move-limit factor approach each other in the later iterations when the move limits have been reduced. In figure 4.8, the constraints computed using formulation 3 initially show a more erratic iteration history than the comparable SLP results. Also evident in this figure is the reduction in the magnitude of this erratic constraint behavior, and the convergence to the constraint boundary as the solution converges.

The number of iterations required for convergence, as shown in figure 4.6, is misleading because the approximate iterations of formulation 3 are so much less expensive than the exact iterations. A more interesting view of these same results is shown in figure 4.9 which uses a CPU time ordinate. The CPU time shown is normalized by the CPU time required to complete the first, exact iteration (this time is about 9 CPU seconds on a SGI IRIS 4D/35 workstation), so using this ordinate is nearly the same as the equivalent number of exact iterations. (The first iteration has some additional set-up logic that makes it longer than a typical exact iteration. The effect of set-up time is noticeable for the ten-bar-truss problem since this problem is small.) Using a

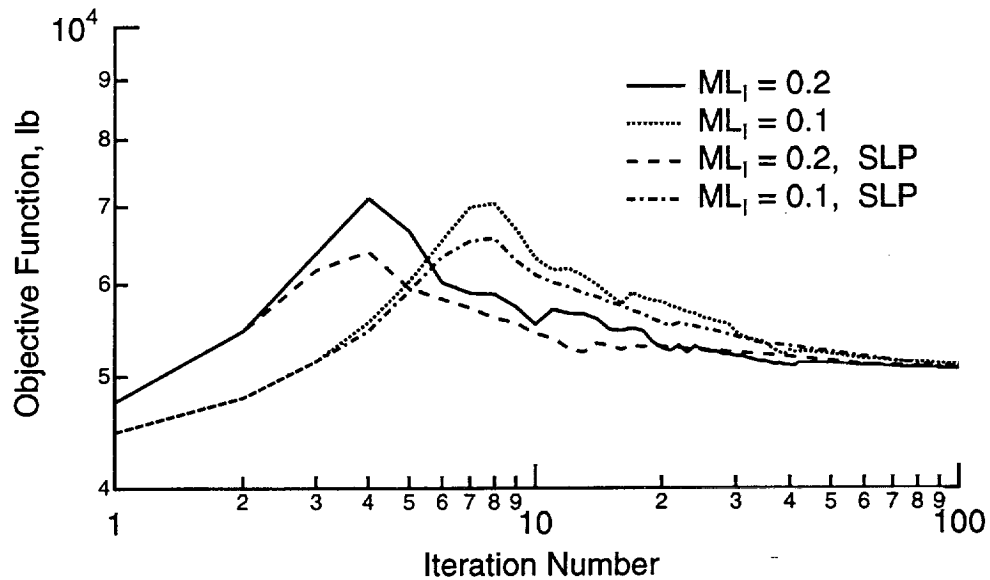


Figure 4.7: Comparison of objective function iteration histories for the ten-bar truss with different initial move-limit-control factors using EP structural-design formulation 3, and using a conventional SLP approach.

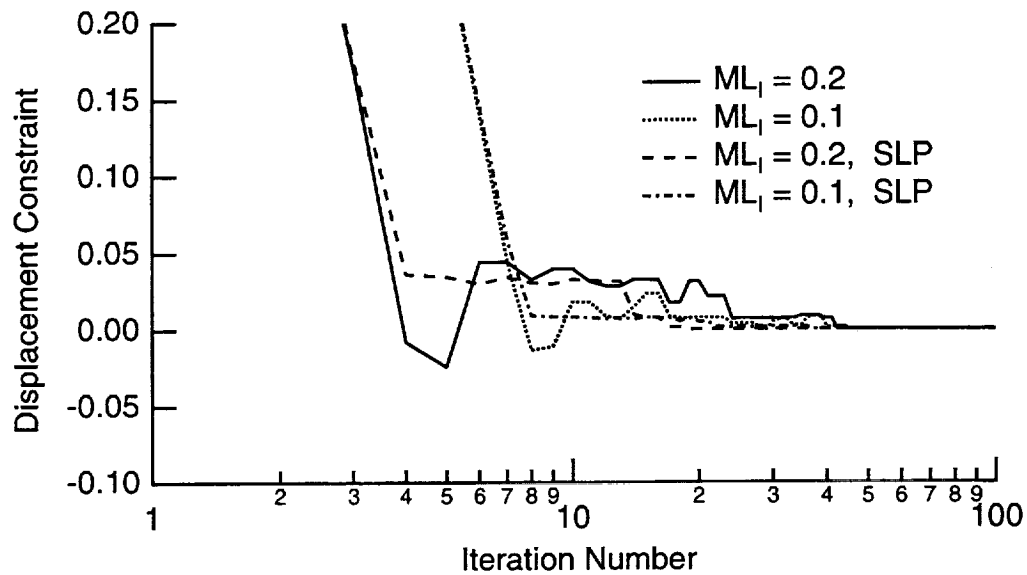


Figure 4.8: Comparison of iteration histories for displacement constraint on δ_1 for the ten-bar truss with different initial move-limit-control factors using EP structural-design formulation 3, and using a conventional SLP approach.

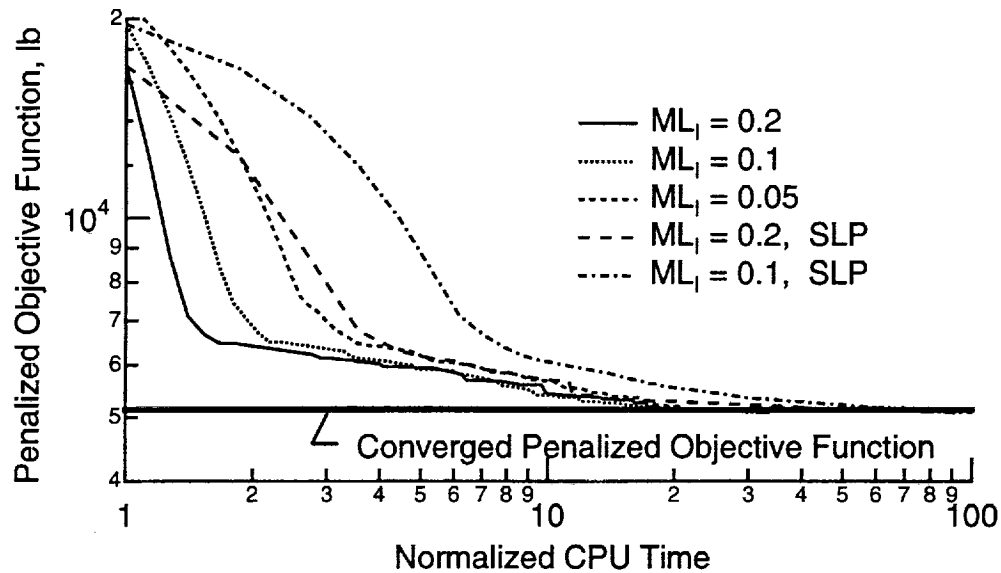


Figure 4.9: Comparison of penalized objective function iteration histories using a CPU time ordinate for the ten-bar truss with different initial move-limit-control factors using EP structural-design formulation 3, and using a conventional SLP approach.

normalized CPU time ordinate, the 20%, 10%, and 5% cases using formulation 3 converge at ordinate values of 21.6, 22.3, and 22.5, respectively. The differences between these convergence times are negligible even though the number of iterations required for convergence is nearly doubled for a 5% initial move-limit case compared to a 20% initial move-limit case. Because the updates to the sensitivity derivatives are more accurate if determined from smaller changes in the design variables, the cases with smaller initial move-limit factors can utilize more approximate iterations before the degradation of the sensitivity derivative accuracy causes an increase in the penalized objective function that signals the need for an exact iteration. However, the smaller the initial move-limit factor, the more iterations that are required to achieve convergence. Thus, using formulation 3, there is a tradeoff in CPU time between the number of

approximate iterations that can be taken between exact iterations, and the total number of iterations. In this case, the reduction in the number of required exact sensitivity analyses, as the initial move-limit factor is decreased, nearly balances the effect of the increased number of iterations on the CPU time required for convergence.

The normalized CPU times required for convergence of the conventional SLP cases for the 20% and 10% initial move-limit factors are 51.1 and 71.0, respectively. In these cases, the benefits of starting with large initial move-limit factors is apparent since all the sensitivity calculations have the same computational cost. These results indicate that for this particular problem, formulation 3 utilizes less CPU time than a conventional SLP formulation by a factor of 2.5 to 3.5.

4.3.2 High-Speed Civil Transport Wing Example Problem Results

The CPU time required for one approximate iteration for the high-speed civil transport wing problem is approximately 8.5% of the time required for an exact iteration, and is only about 1% larger than the cost of performing the structural response analysis and constraint evaluation alone. The former value is a reduction of relative CPU time to nearly half of the value for the ten-bar-truss problem, and the reduction is primarily due to the increase in the number of design variables for the wing problem. The latter value indicates that the constraint update and optimization steps in an approximate iteration are minor in computational cost compared to the structural response analysis and constraint evaluation.

The convergence histories for the penalized objective function for five

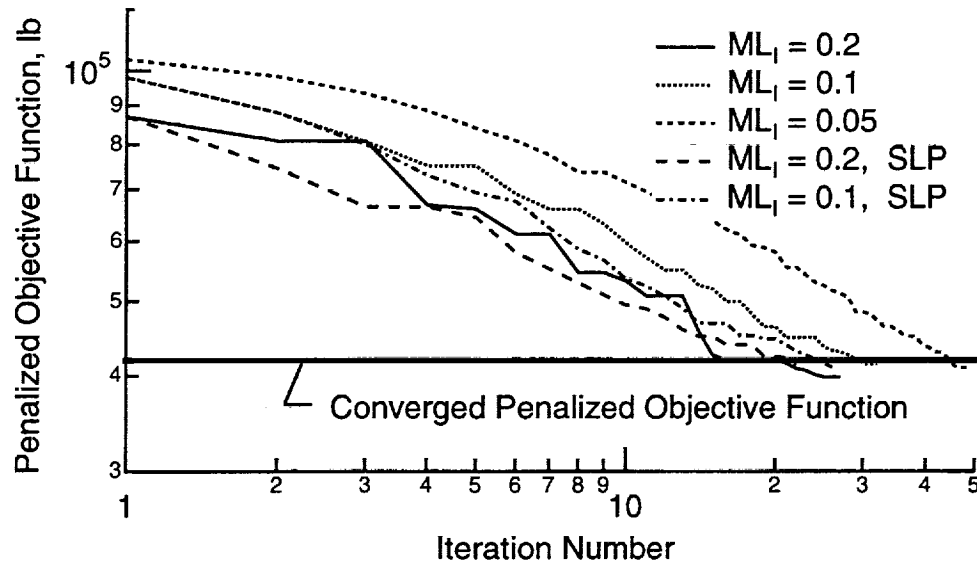


Figure 4.10: Comparison of penalized objective function iteration history for the high-speed civil transport wing with different initial move-limit-control factors using EP structural-design formulation 3, and using a conventional SLP approach.

different cases are shown in figure 4.10. In this figure, results are shown using formulation 3 with initial move-limit factors of 20%, 10%, and 5%. Results using a conventional SLP solution approach with initial move-limit factors of 20% and 10% are also shown for comparison. As in the ten-bar-truss problem, the solution procedure is halted when the move-limit factor is reduced below 1%. However, there is greater variation in the final penalized objective function among the design cases for this problem. In particular, the final value for the 20% case using formulation 3 is somewhat lower than the other cases. Because of these variations, a penalized objective function value 1% higher than the final value for the SLP case with a 10% initial move-limit factor is defined to be the criterion for convergence. This penalized objective function value is the largest final value for the five cases, and is shown by the horizontal line

in figure 4.10. The 20%, 10%, and 5% cases using formulation 3 converge after 20, 30, and 45 iterations, respectively. However, the final value for the 20% case is achieved at iteration 27 at an appreciably lower value than the “converged” value. The 20% and 10% SLP cases converge at iterations 22 and 25, respectively. Both the number of iterations required for convergence, and the increase in the number of iterations required for convergence as the initial move-limit factor is decreased are much smaller than for the ten-bar-truss problem. The penalized objective function history for SLP using a 20% initial move-limit factor has lower values than the comparable formulation 3 history except for a surprisingly rapid reduction in the penalized objective function for formulation 3 beyond iteration 12. The more rapid convergence of formulation 3 during the final iterations than the SLP approach is believed to be a function of the different solution paths in design space having no general significance. The 10% initial move-limit-factor cases yield results that are consistent with expectations. Here the solution paths of the formulation 3 and the SLP cases follow each other closely for the initial iterations before they diverge due to an accumulation of errors in the sensitivity derivatives. The penalized objective function for the SLP case is less than the value using formulation 3 throughout the design history after the paths diverge. The number of exact sensitivity calculations required for convergence for the 20%, 10%, and 5% cases using formulation 3 are six, seven, and eight, respectively. Thus, the ratio of the number of approximate iterations to the number exact iterations increases from 2.3 for the 20% case to 4.6 for the 5% case because, as in the ten-bar-truss problem, the accuracy of the updates is greater for the smaller initial move-limit factors.

The iteration histories of the objective function and the displacement constraint on the wing tip deflection are shown in figure 4.11 for the cases having an initial move-limit factor of 10%. As seen in the figure, the objective function values generally decrease to the minimum weight design since the initial design is feasible. The objective function results show that formulation 3 and the SLP results agree well for the initial iterations, when the displacement constraint is not active. Then from iterations 5 through 18, objective function for formulation 3 decreases at a slightly lower average rate than the SLP objective function, partially due to the inclusion of results from rejected iterations. The displacement constraint iteration history for formulation 3 shows initial satisfaction of the constraint, a gross violation of the constraint for iterations 4 and 5, and a fairly close tracking of the constraint boundary for the final iterations. Thus, the convergence of the objective function and the displacement constraint are seen to be only moderately affected by the use of approximate constraint sensitivity derivatives.

The penalized objective function results for this problem are shown in figure 4.12 using a CPU time ordinate. The CPU time shown is normalized by the CPU time required to complete the first, exact iteration (this time is about 1830 CPU seconds on a SGI IRIS 4D/35 workstation). Using the normalized CPU time ordinate, the 20%, 10%, and 5% cases using EP structural-design formulation 3 converge at ordinate values of 6.3, 9.6, and 11.6, respectively. Here, the effect of the initial move-limit factor on the CPU time favors the large initial move-limit factors simply because there are fewer exact iterations for those cases. The normalized CPU times required for convergence of the conventional SLP cases for the 20% and 10% initial move-limit factors are 18.5

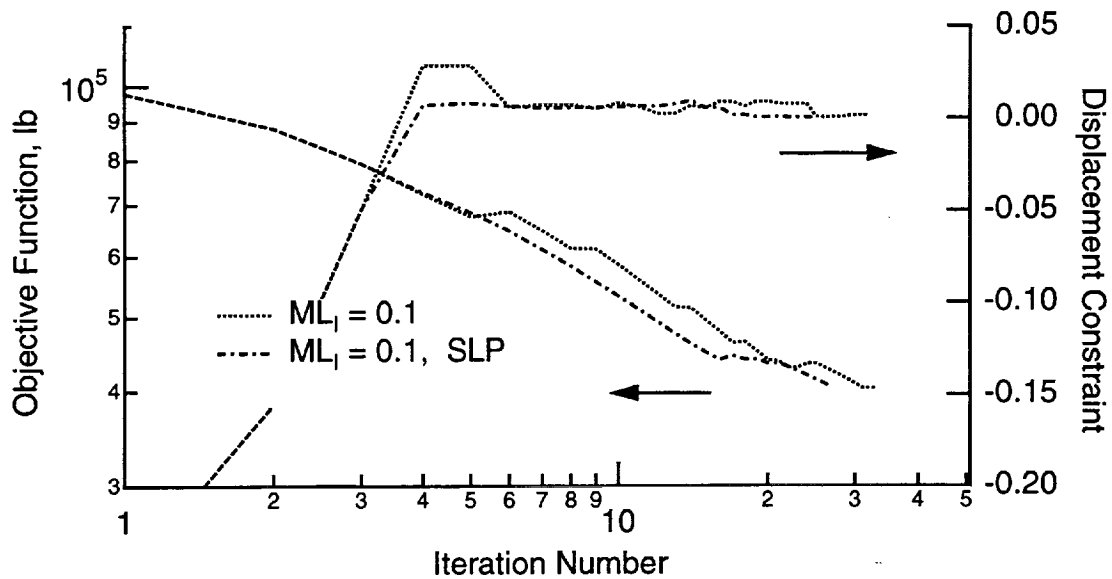


Figure 4.11: Objective function and wing tip displacement constraint iteration histories for the high-speed civil transport wing using EP structural-design formulation 3, and using a conventional SLP approach.

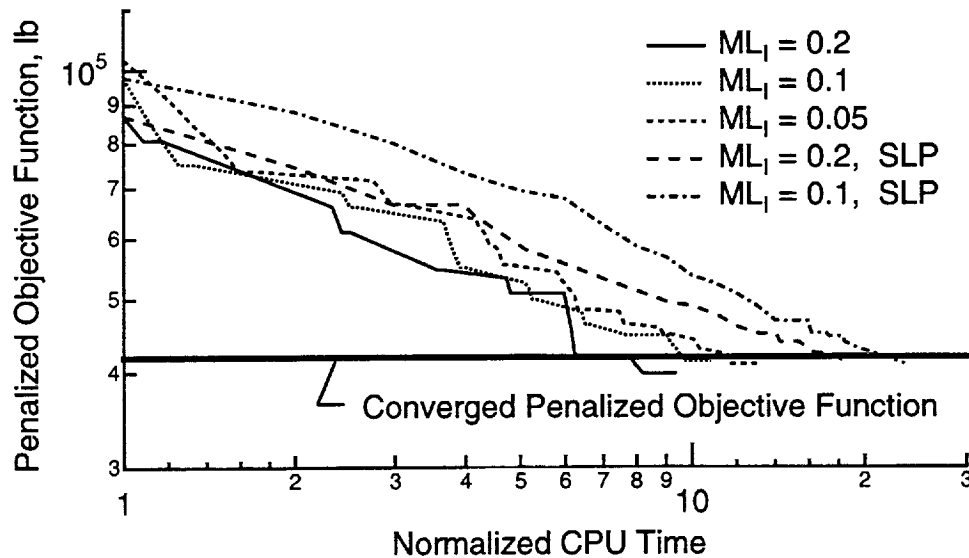


Figure 4.12: Comparison of penalized objective function iteration history using a CPU time ordinate for the high-speed civil transport wing with different initial move-limit-control factors using EP structural-design formulation 3, and using a conventional SLP approach.

and 22.3, respectively. These results indicate that for this wing design problem, formulation 3 requires less CPU time than conventional SLP formulations by about a factor of 2.

4.3.3 High-Speed Transport Vehicle Example Problem Results

The CPU time required for one approximate iteration for the high-speed transport vehicle problem is approximately 1.5% of the time required for an exact iteration. Thus because of the large number of design variables, the cost of an approximate iteration for this problem is insignificant compared to the cost of an exact iteration. The cost of an approximate iteration is only about 2% larger than the cost of performing the structural response analysis and constraint evaluation alone. This slight increase from the previous wing example is possibly due to the larger number of design variables in the optimization step for this problem, but may also be an effect of the use of different types of computers for solving these two cases.

The convergence histories for the penalized objective function, normalized by its initial value, for three different cases are shown in figure 4.13. All results for this problem are obtained utilizing an initial move-limit factor of 10% because the computational cost of this problem prohibited investigation of results for different initial move-limit factors. The three cases shown in the figure are labeled formulation 3, modified formulation 3, and SLP. The need for and the definition of the modified formulation 3 requires further explanation. The final designs (i.e., the sizing-variable values when the move-limit factor is reduced below 1%) for the formulation 3 and SLP cases are appreciably different in figure 4.13. These differences in the final designs makes it difficult to

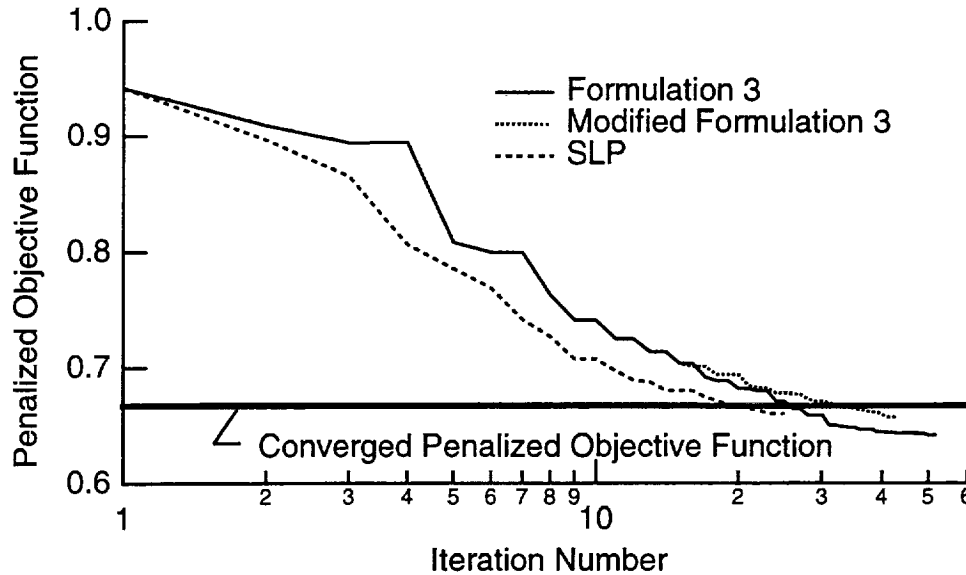


Figure 4.13: Comparison of penalized objective function iteration history for the high-speed transport vehicle using EP structural-design formulation 3, using modified EP structural-design formulation 3, and using a conventional SLP approach.

compare the convergence characteristics of the two approaches. For instance, when using a convergence definition as in the previous example problems (i.e., when the penalized objective function is at a value 1% higher than the final value of the SLP case), the move-limit factor at convergence for the formulation 3 results is 10% while the move-limit factor for SLP is 2.5%. In addition, results to be shown subsequently demonstrate significant constraint violations for formulation 3 when using this convergence definition. These difficulties in comparing convergence characteristics are overcome by using a less conservative move-limit-factor reduction strategy, namely the modified formulation 3. In the modified formulation 3, the move-limit-factor control of formulation 3 is retained except that, if the first approximate iteration following an exact iteration would increase the penalized objective function, the design is reset to the

values existing before the iteration, and the approximate iteration is repeated with the move-limit factor reduced by half. Using the modified formulation 3, the move-limit factor is 1.25% and the constraints are better satisfied at convergence. The final designs for the modified formulation 3 and SLP cases are much closer, as demonstrated in figure 4.13, which allows meaningful direct comparisons between these two methods.

In figure 4.13, formulation 3 “converges” in 26 iterations, modified formulation 3 converges in 34 iterations, and the SLP case converges in 21 iterations. However, as mentioned previously, the present definition of convergence is not appropriate for formulation 3 which has a final value for the penalized objective function that is much lower than the final value for the modified formulation 3 and the SLP cases. The solution paths of the formulation 3 and the modified formulation 3 cases follow each other closely for the initial iterations before they diverge due to move limit reductions in the modified formulation. The number of exact sensitivity calculations required for “convergence” for formulation 3 is ten, and for convergence for modified formulation 3 is eleven.

The iteration histories of the objective function and the most critical stress constraint (i.e., the constraint having the largest Lagrange multiplier) are shown in figure 4.14 for the cases using EP structural-design formulation 3, using modified EP structural-design formulation 3, and using a conventional SLP approach. As seen in figure 4.14, the objective function values for all three cases agree prior to iteration 5, and beyond this iteration number the formulation 3 objective function values follow a more erratic path due to errors in the sensitivity derivatives that lead to rejected iterations. The two formulation 3 cases diverge from a common path at iteration 12 where the move-limit factor

for the modified formulation is decreased. The final value for the objective function is appreciably lower for formulation 3 case indicating a different final, and more optimal, design than the other cases. The crossover of the objective function values for the formulation 3 and the SLP cases that occurs near iteration 18 in figure 4.14 does not occur for the penalized objective function in figure 4.13 because the constraint histories using formulation 3 have larger violations than the SLP case. These constraint violations for the formulation 3 results are due to the 10% move-limit factor being retained in all iterations up to “convergence”, and are typified by the stress constraint in figure 4.14. The stress constraints for modified formulation 3 and SLP in this figure have small, but not insignificant, violations at the iteration numbers where the solutions are said to be converged, and the violation is nearly 10% when the penalized objective function for formulation 3 achieves the “converged” value in figure 4.13.

The penalized objective function results shown in figure 4.13 are repeated in figure 4.15 using a CPU time ordinate. The CPU time shown is normalized by the CPU time required to complete the first, exact iteration (this time is about 7200 CPU seconds on a Digital DEC 3000 Model 500 workstation). Using the normalized CPU time ordinate, formulation 3 “converges” at an ordinate value of 10.4, modified formulation 3 converges at 11.5, and the conventional SLP approach converges at 19.1. Because of the insignificant cost of the approximate iterations in the EP formulations for this problem, the normalized CPU time essentially measures the cost of performing the exact sensitivity analyses. In spite of the large difference in the number of iterations to convergence, there are only small differences in computational effort between

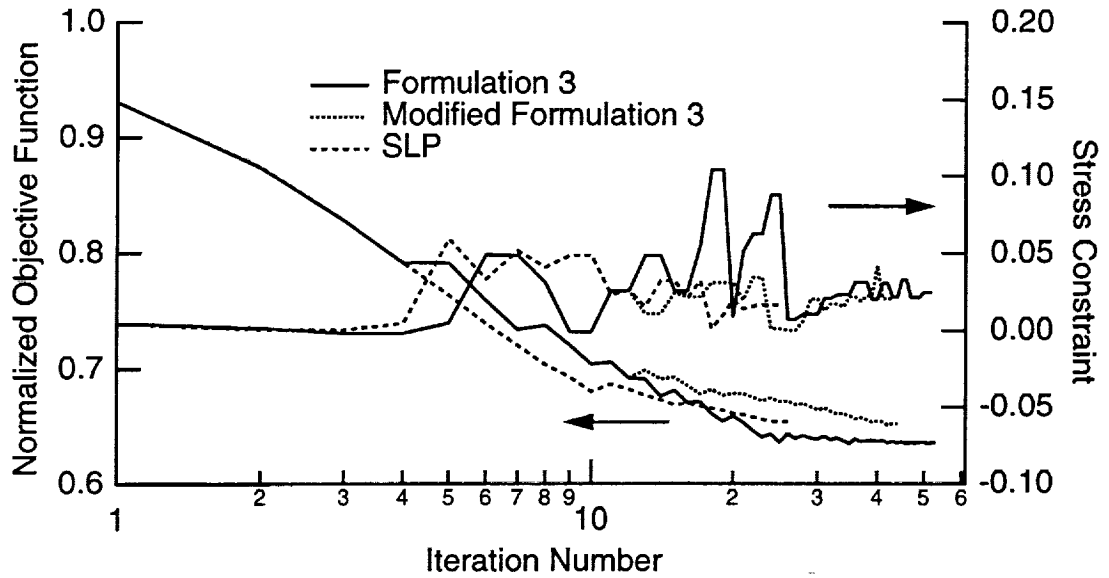


Figure 4.14: Comparison of objective function and most critical stress constraint iteration history for the high-speed transport vehicle using EP structural-design formulation 3, using modified EP structural-design formulation 3, and using a conventional SLP approach.

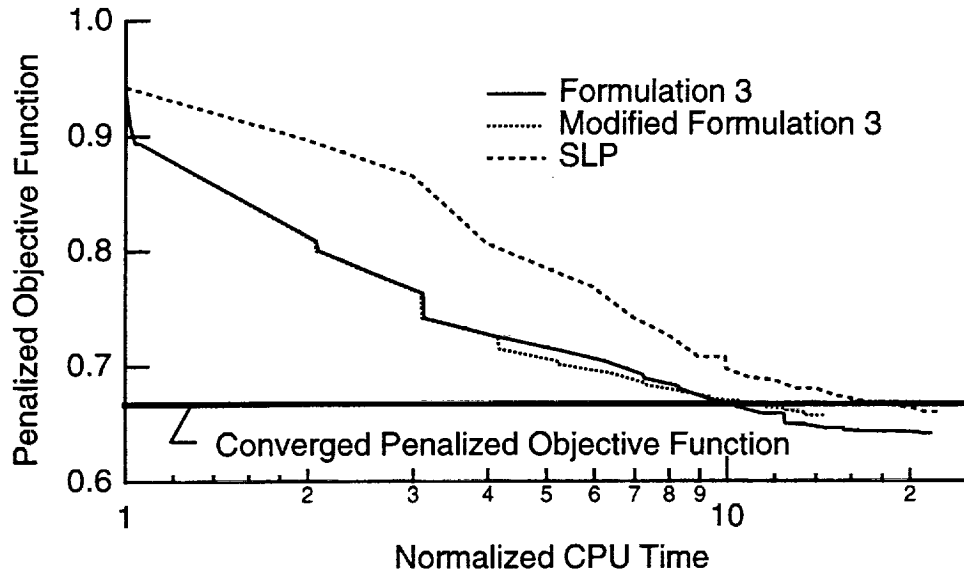


Figure 4.15: Comparison of penalized objective function iteration history using a CPU time ordinate for the high-speed transport vehicle using EP structural-design formulation 3, using modified EP structural-design formulation 3, and using a conventional SLP approach.

formulation 3 and the modified formulation 3 for this problem. Comparing the modified formulation 3 and the SLP cases for this vehicle design problem, modified formulation 3 utilizes less CPU time than conventional SLP formulations by a factor of nearly 1.7.

4.3.4 Transmission Tower Example Problem Results

In this subsection, results are presented for the minimum weight design of the transmission tower problem using EP structural design formulation 4 with decompositions into two and four substructures. Results from the SLP solution for the entire tower structure are also presented. No move limits are utilized for the interface force and the rigid-body mode design variables in the structural-sizing subproblem for formulation 4 results. The initial move-limit factor for the sizing design variables is 10% for all cases. There are only minor differences between the iteration histories for the objective function and the penalized objective function, so only the former is described. The iteration histories for the objective function and the most critical stress constraint (i.e., for the vertical member with maximum compression in the lowest bay) are shown in figure 4.16 using the three partitions of the tower into substructures. The three cases are indistinguishable in the figure, and convergence to a weight that is 1% larger than the final value of 1212 lb occurs by iteration 42. The most critical stress constraint value increases to overshoot the constraint boundary by iteration 9, but rapidly recovers to track closely the constraint boundary thereafter.

The compatibility and orthogonality equality constraints in structural-sizing subproblem (3.38) have been found to converge rapidly for this example

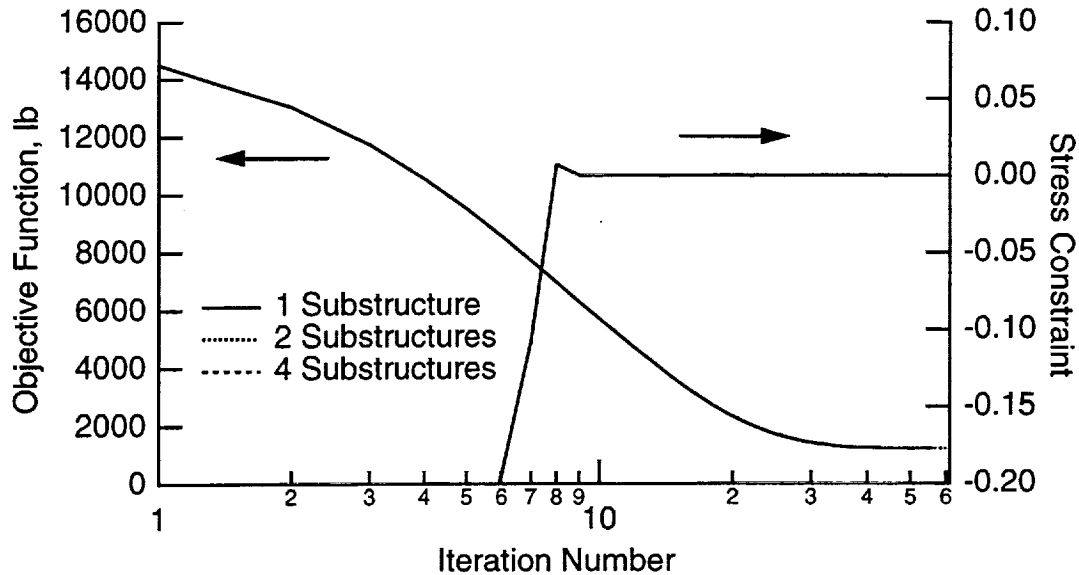


Figure 4.16: Comparison of objective function and most critical stress constraint iteration history for the transmission tower example using EP structural-design formulation 4 with two and four substructures, and using a conventional SLP approach for the entire structure.

problem. After 2 iterations, the residuals of the interface displacement compatibility constraints are below 0.026 in. for the two substructure partitioning, and below 0.016 in. for the four substructure partitioning. There is some oscillation in the value of these residuals during the solution process, but at convergence the residuals are below 2.8×10^{-5} in. and 1.7×10^{-4} in. for the two and four substructure cases, respectively. The orthogonality equality constraints in subproblem (3.38) are essentially satisfied after the first iteration, with residuals of order 10^{-10} lb or smaller throughout the solution process.

A summary of the CPU timing results are shown in table 4.1. In this table, the three columns denoted by 1 substr., 2 substr., and 4 substr. show results for the SLP case, and the two and the four substructure cases using formulation 4, respectively. The CPU times for analyses of the structural re-

	1 Substr.	2 Substr.	4 Substr.
Number Equality Constraints	0	18	54
Substructure ID	All	1	I
Number Sizing Variables	279	140	82
Analysis CPU Time	0.0130	0.0050	0.0027
Sensitivity CPU Time	0.9822	0.2136	0.0721
Substructure ID		2	II
Number Sizing Variables		139	58
Analysis CPU Time		0.0051	0.0027
Sensitivity CPU Time		0.2120	0.0443
Substructure ID			III
Number Sizing Variables			74
Analysis CPU Time			0.0033
Sensitivity CPU Time			0.0660
Substructure ID			IV
Number Sizing Variables			65
Analysis CPU Time			0.0022
Sensitivity CPU Time			0.0485
Optimization CPU Time	0.0028	0.0025	0.0038
Misc. CPU Time	0.0020	0.0010	0.0037
Total Serial CPU Time	1.0000	0.4391	0.2495
Estimated Parallel CPU Time	1.0000	0.2206	0.0823

Table 4.1: Comparison of normalized CPU timing results for the transmission tower example using EP structural-design formulation 4 with two and four substructures, and using a conventional SLP approach for the entire structure.

sponses that include evaluations of the substructure constraints, and for the sensitivity derivative evaluations within each substructure are shown. CPU times are also shown for solving structural-sizing subproblem (3.38), which optimizes the structural design and determines the coordination inputs, as well as for performing miscellaneous computations. All CPU times are normalized by the total CPU time required for the SLP case. The time for one iteration of the SLP case is 2840 CPU seconds on a SGI IRIS 4D/35 workstation. The next to last row shows the CPU time requirements for the solution of the tower example problem using the serial approach of the present dissertation, and the last row contains estimates of the CPU time requirement of the longest computational thread if the substructure analysis and sensitivity derivative calculations were done in parallel. These latter estimates simply combine the largest substructure analysis and sensitivity CPU time values in table 4.1 with the miscellaneous and optimization CPU time values.

As seen in table 4.1, even in the serial mode this example is well suited to substructuring. Utilizing two substructures reduces the CPU time to less than 50% of the SLP value, and utilizing four substructures reduces the CPU time to less than 25% of the SLP value. The parallel processing estimates reduce these values to 22% and 8.2%, respectively. Thus, the computational benefits of the substructuring formulation can be appreciable. The number of coordination variables and the number of related coordination equality constraints increases with the number of substructures, and this increase is reflected in the increase in the optimization CPU time in going from two to four substructures. However, the optimization CPU time for this problem is not large for any of the three cases.

Chapter 5

Concluding Remarks

The overall goal of the present dissertation is the development of finite-element-based, optimal, structural-sizing methods to solve large-scale problems more efficiently than the current, commonly used methods. The approach taken in the development of the new structural-sizing methods is to base them on equilibrium programming (EP) formulations to take advantage of the theory that exists in EP. A review of the commonly used methods indicated that the most fruitful approach would be equilibrium programming formulations that reduce the cost of sensitivity derivative calculations. So this approach was utilized in the development of the formulations.

To acquaint the reader with equilibrium programming, background information was presented to describe the history of equilibrium programming, to define an equilibrium programming problem, and to summarize conditions necessary and sufficient for the existence of an equilibrium point. Properties that distinguish an equilibrium point and an optimal point, and various solution methodologies were also described.

Four equilibrium programming, structural-design formulations were then developed. In developing these formulations, the implications of the necessary conditions and the constraint qualification on solution existence were utilized.

The first two formulations, namely formulations 1 and 2, are of interest because they are essentially the commonly used methods of fully stressed design, and of using rapid, approximate analyses in a nonlinear-programming-based structural design method. The former requires no sensitivity derivatives of the design constraints, and the latter uses sensitivity derivatives sparingly to define the approximate analyses. Thus, two commonly used design methods can be viewed as EP formulations. Two additional EP structural design formulations, formulations 3 and 4, are also derived in the present dissertation. In formulation 3, additional EP subproblems define inexpensive, approximate sensitivity derivative updates that make formulation 2 even more efficient. In formulation 4, a decomposition into EP subproblems is derived in which the structural response and sensitivity derivative calculations of individual substructures are decoupled. For structures amenable to substructuring, the computational effort in calculating sensitivity derivatives in the substructure EP subproblems is greatly reduced from what would be required to calculate them using a single large structure. Thus, all four formulations faithfully follow the approach of reducing either the number or the size of the sensitivity derivative calculations required in the structural design.

Algorithms were developed to implement the two new EP formulations by utilizing a commercial finite-element analysis package. Another algorithm that is based on sequential linear programming (SLP) methods was used to define a commonly used method that is used as a basis for comparison. The algorithm for formulation 3 is dependent on the penalized objective function for determining the acceptability of using sensitivity derivative updates, and for determining the move-limit factor reductions. This algorithm was applied

to three example problems, ranging from a simple ten-bar-truss problem to a large vehicle optimization problem. As expected, the results showed that the computational cost of the exact sensitivity derivative calculations dominates the overall cost, even for the small ten-bar-truss problem. The sensitivity derivative updates were shown to be most effective on small problems, reducing the computational cost of obtaining a converged solution by factors ranging from 2.5 to 3.5. However, the sensitivity derivative updates also were useful on a large problem with nearly 350 design variables, reducing the computational cost by a factor of nearly 1.7.

An algorithm implementing the substructure-based, formulation 4 was also developed and applied to a transmission tower with nearly 280 design variables. This example has relatively few interface nodes between the substructures, and thus was an ideal example for this formulation. The iteration history for the weight and the most critical constraint using this formulation with two partitionings of the structure were indistinguishable from the results using the SLP approach. This result was surprising since the interface compatibility constraints are not necessarily satisfied until the design converges. The CPU time for the sensitivity derivative calculations using the SLP approach was over 98% of the total CPU time, and this time was reduced by over 75% when using four substructures. Thus, the substructure-based formulation can be very effective in reducing the computational time. An approach combining sensitivity derivative updates with the substructure-based formulation was not attempted, but may show excellent promise to reduce computational time by combining the best features of both formulations.

The development of structural design formulations based on equilibrium

programming, described herein, has been very successful. Although equilibrium programming theory did not directly define the new formulations, the success of their development owes much to the recognition and utilization of the conditions for EP solution existence. Thus, EP is really only a framework that is used as a setting for development of new design methods. The developer still has the responsibility for identifying the sources of inefficiency in present approaches, developing formulations that address these inefficiencies, and ensuring that the formulations developed are at least locally optimal. Another benefit of the equilibrium programming framework is that the developer gains the new perspective of viewing the design process as consisting of a number of interacting subproblems, each pursuing their own goals and in need of coordination for development of satisfactory designs. Further enhancements to computational efficiency by exploiting parallel solutions of these subproblems are implied in the present dissertation, but no parallel implementations were attempted. The work on asynchronous solution of equilibrium programming problems cited in the literature may prove useful in exploiting parallel computation. However, much additional work is necessary for the development of viable parallel solution schemes.

Bibliography

- [1] Arora, J. S., and Govil, A. K.: An Efficient Method for Optimal Structural Design by Substructuring. *Computers and Structures*, Vol. 7, 1977, pp. 507-515.
- [2] Barthelemy, J.-F. M., and Haftka, R. T.: Approximation Concepts for Optimum Structural Design – A Review. *Structural Optimization*, Vol. 5, 1993, pp. 129-144.
- [3] Barthelemy, J.-F. M., and Riley, K. M.: Improved Multilevel Optimization Approach for the Design of Complex Engineering Systems. *AIAA Journal*, Vol. 26, No. 3, March 1988, pp. 353-360.
- [4] Barthelemy, J.-F. M., and Sobieszczanski-Sobieski, J.: Optimal Sensitivity Derivatives of Optimum Objective Functions in Nonlinear Programming. *AIAA Journal*, Vol. 21, No. 6, June 1983, pp. 913-915.
- [5] Barthelmy, J.-F. M.: Engineering Design Applications of Multilevel Optimization Methods. *Proceedings of the First International Conference on Computer Aided Optimum Design of Structures: Applications*, Southampton, UK, June 1989, pp. 113-122.
- [6] Basar, T.: Relaxation Techniques and Asynchronous Algorithms for On-Line Computation of Noncooperative Equilibria. In *Proceedings of the*

- 26th IEEE Conference on Decision and Control* Los Angeles, CA, December 1987, pp. 275-280.
- [7] Bertsekas, D. P.: Convexification Procedures and Decomposition Methods for Nonconvex Optimization Problems. *Journal of Optimization Theory and Applications*, Vol. 29, No. 2, October 1979, pp. 169-197.
 - [8] Bloebaum, C. L.: *Formal and Heuristic System Decomposition Methods in Multidisciplinary Synthesis*. NASA CR-4413, December 1991.
 - [9] Camarda, C. J., and Adelman, H. M.: *Static and Dynamic Structural-Sensitivity Derivative Calculations in the Finite-Element-Based Engineering Analysis Language (EAL) System*. NASA TM-85743, March, 1984.
 - [10] Debreu, G.: A Social Equilibrium Existence Theorem. *Proceedings of the National Academy of Science*, Vol. 38, 1952, pp. 886-893.
 - [11] Dennis, J. E., and Schnabel, R. B.: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.
 - [12] Farhat, C., and Roux, F.-X.: A Method of Finite Element Tearing and Interconnecting and Its Parallel Solution Algorithm. *International Journal for Numerical Methods in Engineering*, Vol. 32, 1991, pp. 1205-1227.
 - [13] Farhat, C.: Fast Structural Design and Analysis Via Hybrid Domain Decomposition on Massively Parallel Processors. *Journal of Computing Systems in Engineering*, Vol. 4, No. 4-6, 1994, pp. 453-472.

- [14] Fox, R. L.: Constraint Surface Normals for Structural Synthesis Techniques. *AIAA Journal*, Vol. 3, No. 8, August 1965, pp. 1516-1517.
- [15] Garcia, C. B., and Zangwill, W. I.: *Pathways to Solutions, Fixed Points, and Equilibria*. Prentice-Hall, 1981,
- [16] Giles, G. L.: Procedure for Automating Aircraft Wing Structural Design. *Journal of the Structural Division, ASCE*, Vol. 97, No. ST1, January 1971, pp. 99-113.
- [17] Ha, C. D.: A Decomposition Method and its Application to Block Angular Linear Programs. *MRC Technical Summary Report 2174*, Mathematics Research Center, University of Wisconsin - Madison, January 1981.
- [18] Ha, C. D.: An Algorithm for Structured, Large-Scale Quadratic Programming Problems. *MRC Technical Summary Report 2276*, Mathematics Research Center, University of Wisconsin - Madison, September, 1981.
- [19] Haftka, R. T., and Gürdal, Z.: *Elements of Structural Optimization, 3rd edition*, Kluwer Academic Publishers, Boston, MA, 1992.
- [20] Haftka, R. T., and Kamat, M. P.: Simultaneous nonlinear structural analysis and design. *Computational Mechanics*, Vol. 4, 1989, pp. 409-416.
- [21] Haftka, R. T.: An Improved Computational Approach for Multilevel Optimum Design. *Journal of Structural Mechanics*, Vol. 12, No. 2, 1984, pp. 245-261.
- [22] Haftka, R. T.: Simultaneous Analysis and Design. *AIAA Journal*, Vol. 23, No. 7, July 1985, pp. 1099-1113.

- [23] Kaneko, I., and Ha, C. D.: A Decomposition Procedure for Large-Scale Optimum Plastic Design Problems. *International Journal for Numerical Methods in Engineering*, Vol. 19, 1983, pp. 873-889.
- [24] Kirsch, U., and Moses, F.: Decomposition in Optimum Structural Design. *Journal of the Structural Division, ASCE*, Vol. 105, No. ST1, January 1979, pp. 85-100.
- [25] Kirsch, U., Reiss, M., and Shamir, U.: Optimal Design by Partitioning into Substructures. *Journal of the Structural Division, ASCE*, Vol. 98, No. ST1, January 1972, pp. 249-267.
- [26] Kirsch, U.: Multilevel Approach to Optimal Structural Design. *Journal of the Structural Division, ASCE*, Vol. 101, No. ST4, April 1975, pp. 957-974.
- [27] Kreisselmeier, G., and Steinhauser, R.: Systematic Control Design by Optimizing a Vector Performance Index. *Proceedings of IFAC Symposium on Computer-Aided Design of Control Systems*, Zurich, 1979, pp. 113-117.
- [28] Lansing, W., Dwyer, W., Emerton, R., and Ranalli, E.: Application of Fully Stressed Design Procedures to Wing and Empennage Structures. *Journal of Aircraft*, Vol. 8, No. 9, 1971, pp. 683-688.
- [29] Lasdon, L. S.: *Optimization Theory for Large Systems*. MacMillan Company, New York, NY, 1970.
- [30] Li, S., and Basar, T.: Distributed Algorithms for the Computation of Noncooperative Equilibria. *Automatica*, Vol. 23, No. 4, 1987, pp. 523-533.

- [31] Lu, P.: Use of Approximate Gradients in Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 5, 1992, pp. 1299-1301.
- [32] Mesarovick, M. D., Macko, D., and Takahara, Y.: *Theory of Hierarchical, Multilevel, Systems*. Academic Press, New York, NY, 1970.
- [33] Murtagh, B. A., and Saunders, M. A.: Minos 5.4 Users's Guide (preliminary). *Technical Report SOL 83-20R*, Department of Operations Research, Stanford University, March 1993.
- [34] Nash, J. F.: Equilibrium Points in N-Person Games. *Proceedings of the National Academy of Science*, Vol. 36, 1950, pp. 48-49.
- [35] Nguyen, D. T.: Multilevel Substructuring Sensitivity Analysis. *Computers and Structures*, Vol. 25, No. 2, 1987, pp. 191-202.
- [36] Noor, A. K., and Lowder, H. E.: Approximate Reanalysis Techniques with Substructuring. *Journal of the Structural Division, ASCE*, Vol. 101, No. ST8, August 1975, pp. 1687-1698.
- [37] Noor, A. K., Kamel, H. A., and Fulton, R. E.: Substructuring Techniques – Status and Projections. *Computers and Structures*, Vol. 8, May 1978, pp. 621-632.
- [38] Padula, S. L., and Polignone, D. A.: New Evidence Favoring Multilevel Decomposition and Optimization. *Third USAF/NASA Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, San Francisco, CA, September 1990, pp. 30-35.

- [39] Pickett, R. M., Rubinstein, M. F., and Nelson, R. B.: Automated Structural Synthesis Using a Reduced Number of Design Coordinates. *AIAA Journal*, Vol. 11, No. 4, April 1973, pp. 489-494.
- [40] Pope, G. G., and Schmit, L. A.: Structural Design Applications of Mathematical Programming Techniques. *AGARD-AG-149-71*, February 1971.
- [41] Rao, S. S., Venkayya, V. B., and Khot, N. S.: Game Theory Approach for the Integrated Design of Structures and Controls. *AIAA Journal*, Vol. 26, No. 4, April 1988, pp. 463-469.
- [42] Rao, S. S.: Game Theory Approach for Multiobjective Structural Optimization. *Computers and Structures*, Vol. 25, No. 1, 1987, pp. 119-127.
- [43] Robins, A. W., Dollyhigh, S. M., Beissner, F. L., Geiselhart, K., Martin, G. L., Shields, E. W., Swanson, E. E., Coen, P. G., and Morris, S. J.: *Concept Development of a Mach 3.0 High-Speed Civil Transport*. NASA TM-4058, September 1988.
- [44] Rockafellar, R. T.: Monotone Operators and the Proximal Point Algorithm. *SIAM Journal of Control and Optimization*, Vol. 14, No. 5, August 1976, pp. 877-898.
- [45] Schmit, L. A., and Farshi, B.: Some Approximation Concepts for Structural Synthesis. *AIAA Journal*, Vol. 12, No. 5, May 1974, pp. 692-699.
- [46] Schmit, L. A., and Fox, R. L.: An Integrated Approach to Structural Synthesis and Analysis. *AIAA Journal*, Vol. 3, No. 6, June 1965, pp. 1104-1112.

- [47] Schmit, L. A., and Merhinfar, M.: Multilevel Optimum Design of Structures with Fiber-Composite Stiffened-Panel Components. *AIAA Journal*, Vol. 20, No. 1, January 1982, pp. 138-147.
- [48] Schmit, L. A., and Miura, H.: *Approximation Concepts for Efficient Structural Synthesis*. NASA CR-2552, March 1976.
- [49] Schmit, L. A., and Ramanathan, R. K.: Multilevel Approach to Minimum Weight Design Including Buckling Constraints. *AIAA Journal*, Vol. 16, No. 2, February 1978, pp. 97-104.
- [50] Schmit, L. A.: Structural Synthesis – Its Genesis and Development. *AIAA Journal*, Vol. 19, No. 10, October 1981, pp. 1249-1263.
- [51] Scotti, S. J.: Structural Design using Equilibrium Programming. *AIAA Paper 92-2365, Proceedings of the 33rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, April 13 - 15, 1992.
- [52] Scotti, S. J.: Structural Design Utilizing Updated Approximate Sensitivity Derivatives. *AIAA Paper No. 93-1531, AIAA/ASME/ASCE/AHS/ASC 34th Structures, Structural Dynamics, and Materials Conference*, April 19-21, 1993.
- [53] Scotti, S. J.: A Structural Design Decomposition Method Utilizing Substructuring. NASA TM 107750, January, 1994.
- [54] Shankar, J., Ribbens, C. J., Haftka, R. T., and Watson, L. T.: Computational Study of a Nonhierarchical Decomposition Algorithm. *Computers and Optimization Applications*, Vol. 2, 1993, pp. 273-293.

- [55] Sobieszczanski-Sobieski, J., Barthelemy, J.-F. M., and Riley, K. M.: Sensitivity of Optimum Solutions to Problems Parameters. *AIAA Journal*, Vol. 20, No. 9, September 1982, pp. 1291-1299.
- [56] Sobieszczanski-Sobieski, J., James, B. B., and Dovi, A.: Structural Optimization by Multi-Level Decomposition. *AIAA Journal*, Vol. 23, No. 11, November 1985, pp. 1775-1782.
- [57] Sobieszczanski-Sobieski, J., James, B. B., and Riley, K. M.: Structural Sizing by Generalized, Multilevel Optimization. *AIAA Journal*, Vol. 25, No. 1, January 1987, pp. 139-145.
- [58] Sobieszczanski-Sobieski, J.: *A Linear Decomposition Method for Large Optimization Problems - Blueprint for Development*. NASA TM-83248, February 1982.
- [59] Sobieszczanski-Sobieski, J.: Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems. *Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, NASA CP-3031, Barthelemy, J.-F. Ed., 1989, pp. 51-78.
- [60] Sobieszczanski-Sobieski, J.: Sensitivity of Complex, Internally Coupled Systems. *AIAA Journal*, Vol. 28, No. 1, January 1990, pp. 153-160.
- [61] Sobieszczanski-Sobieski, J.: Two Alternative Ways for Solving the Coordination Problem in Multilevel Optimization. *NASA TM 104036*, August 1991.
- [62] Stephanopoulos, G., and Westerberg, A. W.: The Use of Hestenes' Method of Multipliers to Resolve Dual Gaps in Engineering System Opti-

- mization. *Journal of Optimization Theory and Applications*, Vol. 15, No. 3, March 1975, pp. 285-309.
- [63] Storaasli, O. O., and Sobieszczanski, J.: On the Accuracy of the Taylor Approximation for Structure Resizing. *AIAA Journal*, Vol. 12, No. 2, February 1974, pp. 231-233.
- [64] Thareja, R. R., and Haftka, R. T.: Efficient Single-Level Solution of Hierarchical Problems in Structural Optimization. *AIAA Journal*, Vol. 28, No. 3, 1990, pp. 506-514.
- [65] Vincent, T. L.: Game Theory as a Design Tool. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 105, June 1983, pp. 165-170.
- [66] Whetstone, W. D.: *EISI-EAL Engineering Analysis Language Reference Manual*, Engineering Information Systems, San Jose, CA., July, 1983.
- [67] Wrenn, G. A., and Dovi, A. R.: Multilevel Decomposition Approach to the Preliminary Sizing of a Transport Aircraft Wing. *Journal of Aircraft*, Vol. 25, No. 7, July 1988, pp. 632-638.
- [68] Zangwill, W. I., and Garcia, C. B.: Equilibrium Programming: The Path Following Approach and Dynamics. *Mathematical Programming*, Vol. 21, 1981, pp. 262-289.
- [69] Zeidler, E.: *Nonlinear Functional Analysis and its Applications III, Variational Methods and Optimization*. Springer-Verlag, New York, 1984.

- [70] Zhang, J., Nae-Heon, K., and Lasdon, L.: An Improved Successive Linear Programming Algorithm. *Management Science*, Vol. 31, No. 10, October 1985, pp. 1312-1331.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 1995		3. REPORT TYPE AND DATES COVERED Technical Memorandum
4. TITLE AND SUBTITLE Structural Design Using Equilibrium Programming Formulations			5. FUNDING NUMBERS WU 537-06-21-09	
6. AUTHOR(S) Stephen J. Scotti				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-0001			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546			10. SPONSORING / MONITORING AGENCY REPORT NUMBER NASA TM-110175	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category - 39, 66			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Solutions to increasingly larger structural optimization problems are desired. However, computational resources are strained to meet this need. New methods will be required to solve increasingly larger problems. The present approaches to solving large-scale problems involve approximations for the constraints of structural optimization problems and/or decomposition of the problem into multiple subproblems that can be solved in parallel. An area of game theory, equilibrium programming (also known as non-cooperative game theory), can be used to unify these existing approaches from a theoretical point of view (considering the existence and optimality of solutions), and be used as a framework for the development of new methods for solving large-scale optimization problems. Equilibrium programming theory is described, and existing design techniques such as fully stressed design and constraint approximations are shown to fit within its framework. Two new structural design formulations are also derived. The first new formulation is another approximation technique which is a general updating scheme for the sensitivity derivatives of design constraints. The second new formulation uses a substructure-based decomposition of the structure for analysis and sensitivity calculations. Significant computational benefits of the new formulations compared with a conventional method are demonstrated.				
14. SUBJECT TERMS Equilibrium programming, optimization, structural design, structural analysis, nonlinear programming			15. NUMBER OF PAGES 115	
			16. PRICE CODE A06	
17. SECURITY CLASSIFICATION OF REPORT Unclassified		18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified		19. SECURITY CLASSIFICATION OF ABSTRACT
				20. LIMITATION OF ABSTRACT

